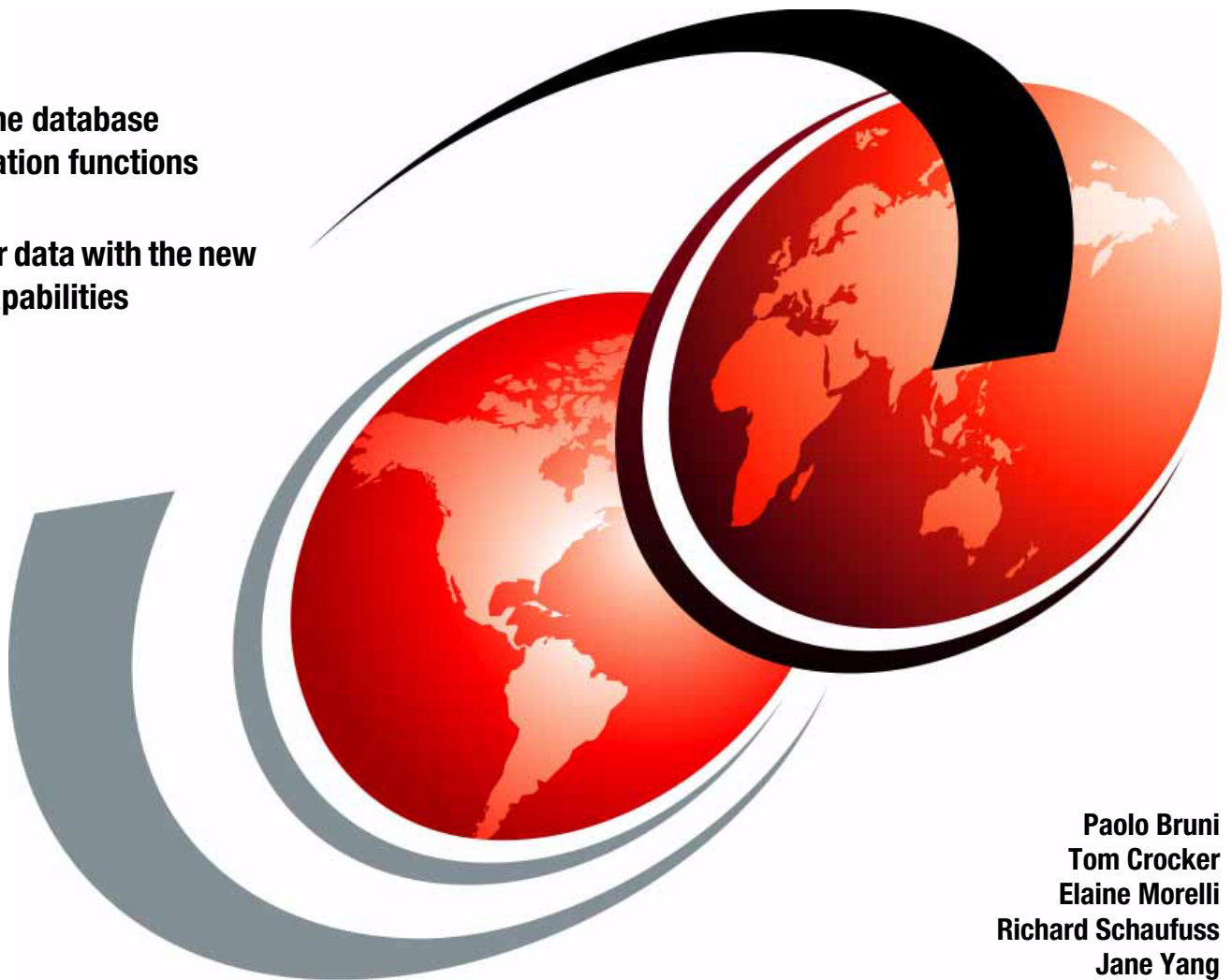


Managing IBM DB2 10 for z/OS Using the IBM DB2 Administration Tool for z/OS Version 10

Use with IBM DB2 10 for z/OS from
Day 1

Simplify the database
administration functions

Adapt your data with the new
change capabilities



Paolo Bruni
Tom Crocker
Elaine Morelli
Richard Schaufuss
Jane Yang

Redbooks



International Technical Support Organization

**Managing IBM DB2 10 for z/OS Using the IBM DB2
Administration Tool for z/OS Version 10**

April 2011

Note: Before using this information and the product it supports, read the information in “Notices” on page xxiii.

First Edition (April 2011)

This edition applies to Version 10, Release 1 of IBM DB2 Administration Tool for z/OS and Version 10.1 of IBM DB2 Object Comparison Tool for z/OS (respectively program number 5655-W34 and 5655-W36) for use with IBM DB2 Version 10.1 for z/OS (program number 5605-DB2).

© Copyright International Business Machines Corporation 2011. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	ix
Tables	xix
Examples	xxi
Notices	xxiii
Trademarks	xxiv
Preface	xxv
The team who wrote this book	xxv
Now you can become a published author, too!	xxvi
Comments welcome.	xxvii
Stay connected to IBM Redbooks	xxvii
Part 1. Introduction	1
Chapter 1. DB2 administration tools at a glance	3
1.1 DB2 Administration Tool for z/OS	4
1.2 DB2 Object Comparison Tool for z/OS	6
1.3 Administration evolution and support for DB2 10 for z/OS	8
Part 2. Installation and customization	11
Chapter 2. Product setup	13
2.1 The underlying components of DB2 Administration Tool	14
2.1.1 Product libraries	14
2.1.2 Checkpoint database	15
2.1.3 Change management database	16
2.1.4 Segmented temporary table space	16
2.1.5 DB2 catalog copy version database (optional)	16
2.1.6 Additional views against DB2 catalog objects (optional)	17
2.1.7 Additional indexes on DB2 catalog objects (optional)	17
2.1.8 Product plans	17
2.1.9 Product collections	18
2.2 Planning the installation of DB2 Administration Tool	18
2.2.1 Organizational resources	18
2.2.2 Product libraries	19
2.2.3 Checkpoint database	21
2.2.4 Segmented temporary table space	22
2.2.5 Change management database (optional)	23
2.2.6 Binding the plans and packages	24
2.2.7 Copying fixed-block to variable-block libraries (optional)	25
2.2.8 Preparing the ADBL CLIST	25
2.2.9 Customizing the operating environment on ISPF	26
2.2.10 Tailoring the DB2 Selection menu (optional)	27
2.2.11 Customizing the ADB2UCUS skeleton	27
2.2.12 Updating the APF authorization table	28
2.2.13 Preparing to run work statement lists online (optional)	28
2.2.14 Creating a catalog copy version database (optional)	29

2.2.15	Creating or updating RUNSTATS views (optional)	30
2.2.16	Granting SELECT access to the DB2 catalog	31
2.2.17	Additional indexes on DB2 catalog objects (optional)	31
2.2.18	Optimizing DSNWZP and DSNZPARM settings	32
2.2.19	Running the RUNSTATS utility against the DB2 catalog	32
2.2.20	Tailoring DB2 Administration Tool authorization switching	33
2.2.21	Enabling DB2 Administration Tool distributed support	33
2.3	Installing a new image of DB2 Administration Tool	34
2.3.1	Product libraries	35
2.3.2	Creating the checkpoint database	37
2.3.3	Creating a segmented temporary table space	40
2.3.4	Change management database	40
2.3.5	Binding the plans and packages	44
2.3.6	Copying fixed-block to variable-block libraries (optional)	46
2.3.7	Preparing the ADBL CLIST	47
2.3.8	Customizing the operating environment on ISPF	48
2.3.9	Tailoring the DB2 Administration Tool main menu	56
2.3.10	Tailoring the DB2 Selection menu	56
2.3.11	Customizing the ADB2UCUS skeleton	57
2.3.12	Updating the APF authorization table	57
2.3.13	Preparing to run work statement lists online	57
2.3.14	Creating a catalog copy version file (optional)	58
2.3.15	Creating and updating RUNSTATS views (optional)	59
2.3.16	Granting SELECT access to the DB2 catalog	60
2.3.17	Creating additional DB2 catalog indexes (optional)	61
2.3.18	Running the RUNSTATS utility against the DB2 catalog (optional)	62
2.3.19	Enabling DB2 Administration Tool distributed support (optional)	62
Chapter 3	Product parameters	65
3.1	Introduction to product parameters	66
3.2	Installation and customization parameters	66
3.3	General operational parameters	70
3.4	Change management ID parameters	77
Part 3	DB2 9 for z/OS and prior support	79
Chapter 4	Native SQL procedures	81
4.1	What a native SQL stored procedure is	82
4.2	Using the CREATE PROCEDURE function to create native SQL stored procedures	82
4.2.1	Create Procedure panel (ADB26CO)	84
4.2.2	CREATE Stored Procedure Parameters (ADB26COU)	85
4.2.3	CREATE Stored Procedure options (ADB26COV)	88
4.2.4	Create Stored Procedure BIND Options (ADB26COW)	89
4.2.5	Create SQL Stored Procedure Body (ADB26COQ)	90
4.3	Managing a native SQL stored procedure	91
4.3.1	ADDV (ADD version) line command	91
4.3.2	Add Procedure (ADB26CO)	92
4.3.3	DRPV versus DROP line command	98
4.3.4	BIND (BIND deploy version) line command	98
Chapter 5	Universal table space	109
5.1	Universal table space	110
5.1.1	Partition-by-growth universal table space	110
5.1.2	Range-partitioned universal table space	110

5.2 DB2 Administration Tool: MAKEPBG	111
5.2.1 MAKEPBG	115
5.2.2 MAKEPBR	129
Chapter 6. CLONE tables	143
6.1 Basic concepts of CLONE tables	144
6.2 DB2 Administration Tool support for CLONE tables	144
6.2.1 Catalog navigation	145
6.2.2 Adding a CLONE table	150
6.2.3 DROP a CLONE table	152
6.2.4 EXCHANGE data	155
6.3 DB2 utilities and CLONE tables	159
6.4 Other DB2 commands and CLONE	162
Part 4. Super processes	163
Chapter 7. The ALT line command	165
7.1 ALT command	166
7.1.1 Making changes in DB2	166
7.1.2 ALT change process	167
7.2 Using ALT	186
7.2.1 Example of adding multiple object types to a change	186
Chapter 8. The MIG line command	213
8.1 What the Migrate command is	214
8.2 Migrate versus Compare	214
8.3 Migrate high level process	215
8.4 Masks	215
8.4.1 Creating a mask	216
8.5 Migration example	222
8.5.1 Scenario	222
Chapter 9. DB2 Object Comparison Tool enhancements	235
9.1 DB2 Object Comparison Tool	236
9.1.1 Defining the source or target	236
9.1.2 Version Scope	237
9.1.3 Mask	238
9.1.4 Ignore	239
9.2 Compare using Version Scope	239
9.3 Compare: Delta change file	255
Part 5. General DB2 Administration Tool functions	257
Chapter 10. Work statement lists	259
10.1 Introduction to work statement lists	260
10.2 Creating work statement lists	260
10.3 Accessing work statement lists	262
10.4 Viewing and altering work statement lists	264
10.5 Cloning work statement lists	266
10.6 Interpreting work statement lists	267
10.7 Validating work statement lists	268
10.8 Running work statement lists	268
10.9 Restarting work statement lists	272
10.9.1 Running a work statement list and implicit commits	272
10.9.2 System-controlled restart	274

10.9.3	User-controlled restart.	274
10.9.4	Restart report only	275
10.9.5	An example of restarting a work statement list	275
10.10	Environment variables.	279
10.10.1	Stored environment variables	280
Chapter 11.	Optional features	281
11.1	Connecting to a different DB2.	282
11.1.1	CONNECT	282
11.1.2	SSID	284
11.2	Useful DB2 Administration Tool commands	284
11.2.1	ALL	284
11.2.2	Report	286
11.2.3	STATUS	290
11.2.4	SEARCH	291
11.2.5	DB2	293
11.3	Features you may have missed	294
11.3.1	Using the DB2 Administration Tool panels for SQL	294
11.3.2	Querying the catalog using nonstandard criteria.	298
Part 6.	DB2 10 for z/OS support.	303
Chapter 12.	Temporal tables	305
12.1	Temporal tables and versioning	306
12.2	DB2 Administration Tool support of the temporal table.	306
12.3	Using the CT function to create a bi-temporal table	306
12.3.1	TBLOPTS command.	321
12.4	Using the AL line command to convert a table to a temporal table.	333
12.4.1	CUSTOMER_COVERAGE table	334
12.4.2	ADD column option.	335
12.4.3	ADD PERIOD option.	340
12.4.4	History table	341
12.4.5	Using the DDL line command to create a history table.	342
12.4.6	ADD VERSIONING option	346
12.4.7	BASE line command.	352
12.4.8	DROP VERSIONING option	354
Chapter 13.	Security	357
13.1	Overview of security features	358
13.2	Separation of duties	358
13.3	Revoking authorities	365
13.3.1	Revoking authorities without cascading	367
13.4	Copying authorities	369
13.5	Defining security constructs	372
13.5.1	Columns masks	373
13.5.2	Row permissions.	384
13.5.3	Trusted context	394
13.5.4	Roles.	406
13.6	Auditing profiles.	410
Chapter 14.	Physical design.	417
14.1	INCLUDE column	418
14.1.1	Adding an INCLUDE column using the AL line command	418
14.1.2	Adding an INCLUDE column using the ALT line command	422

14.1.3 Removing an INCLUDE column	432
14.2 Inline LOBs	433
14.2.1 What are Large Objects	433
14.2.2 What are inline LOBs	434
14.2.3 Creating a table with an inline LOB column	434
14.2.4 Determining the length of the inline portion of the LOB column	437
14.2.5 Altering the inline length of the LOB	437
14.3 XML	442
14.3.1 What is XML	442
14.3.2 Creating a table with XML columns.	443
14.3.3 Altering a table to add an XML column with an XML modifier.	444
14.3.4 Creating an index on an XML column.	448
14.4 Adding an active log data set	452
14.5 ADD PARTITION	454
14.6 TIMESTAMP with TIME ZONE	455
14.7 CONCENTRATE STATEMENTS WITH LITERAL	460
14.8 USE CURRENTLY COMMITTED	460
14.9 ALTER BUFFERPOOL PAGESTEAL	461
Chapter 15. Online schema evolution support.	465
15.1 Online schema evolution and DB2 pending changes	466
15.2 DB2 Administration Tool support for DB2 pending changes.	466
15.2.1 Scenario 1: Changes that result in a pending state	470
15.2.2 Scenario 2: Mixing change types	478
Chapter 16. General options	483
16.1 GEN	484
16.1.1 GEN prompt to run SQL	484
16.1.2 GEN and pending changes.	487
16.1.3 GEN statement types	488
16.2 Catalog navigation	489
16.3 Performance queries.	494
Part 7. Appendixes	497
Appendix A. Reference tables	499
DB2 Administration Tool install and upgrade planning worksheet.	500
Mask translation names	504
Migration parameter details	506
Abbreviations and acronyms	509
Related publications	511
IBM Redbooks	511
Other publications	511
Online resources	511
How to get Redbooks.	511
Help from IBM	512
Index	513

Figures

1-1 DB2 Administration Tool overview	4
1-2 DB2 Object Comparison Tool overview	7
2-1 DB2 Administration - Customization main menu.	50
2-2 Example Customization - General Parameters panel.	51
2-3 DB2 subsystem customization main menu.	52
3-1 DB2 Administration - Customization menu	66
3-2 Customization - General Parameters panel	67
3-3 Customization - DB2 Subsystem Parameters panel.	68
3-4 Customization - Copy Parameters panel	69
3-5 DB2 Change DB2 Admin Options panel.	70
3-6 Change Colors and Highlights panel	71
3-7 DSNALTR Analysis Options panel	74
3-8 DSNALTR Batch Job Utility Parameters panel	75
3-9 Change Migrate Options panel	76
3-10 Prompt Options panel	76
3-11 Change Management (CM) panel.	77
3-12 CM - Manage ID Table panel	77
4-1 Option 2.4 from DB2 Administration Menu	83
4-2 Specify CO (Create Stored Procedure) option on ADB26	84
4-3 Create SQL native stored procedure	85
4-4 Enter values for Parameter 1	86
4-5 Enter value for Parameter 2	87
4-6 Enter values for parameter 3	87
4-7 Specify Stored Procedure Options	88
4-8 Enter Create Stored Procedure BIND Options	89
4-9 Enter SQL stored procedure body	90
4-10 SQL stored procedure created	90
4-11 New native stored procedure created.	91
4-12 ADDV line command.	92
4-13 Alter Procedure panel	92
4-14 Modify stored procedure informational panel	93
4-15 Alter stored procedure options for ADDV	93
4-16 Alter stored procedure BIND options for ADDV	94
4-17 The SQL body of the stored procedure to be changed.	95
4-18 Modified SQL body	95
4-19 Alter stored procedure ADD VERSION executed.	96
4-20 ADB21O after ADDV executed successfully.	96
4-21 Activate version.	97
4-22 Newly activated version	97
4-23 BIND deploy stored procedure	98
4-24 Specify BIND deploy options	99
4-25 BIND deploy executed successfully	99
4-26 SSID primary command	100
4-27 Switch over to VA1B subsystem.	101
4-28 DB2 Admin Tool main panel on VA1B system	102
4-29 Search stored procedure name starting with NSP_FIND	103
4-30 Located stored procedure NDP_FIND_MEMID	103
4-31 Issue I line command against stored procedure	104

4-32	Interpretation of stored procedure	105
4-33	Issue K line command to stored procedure	106
4-34	Package NSP_FIND_MEMID displayed	106
4-35	Issue I line command against the package	107
4-36	Interpretation of NSP_FIND_MEM package	107
5-1	Enter option 1 on DB2 Admin main menu	112
5-2	Locate the TPAADMR2 table space	113
5-3	TPAADMR2 table space	113
5-4	Interpret line command	114
5-5	Interpretation of table space TPAADMR2	115
5-6	Issuing the ALT command against TPAADMR2	116
5-7	Issuing the MAKEPBG primary command	117
5-8	ADB21SAR updated with conversion to PBG information	117
5-9	Issuing the CONTINUE command to continue the MAKEPBG processing	118
5-10	ADB27CA is the central hub of ALT processing	119
5-11	Enter the ALTER command on ADB27CA	120
5-12	Specify options for building analyze and apply jobs	121
5-13	Create the Apply JCL data set	122
5-14	Analyze job of the ALT of TPAADMR2	123
5-15	Enter ? next to the ALT analyze job	124
5-16	Enter S next to the REPORT	124
5-17	Compare REPORT of the ALT analyze job	125
5-18	Select option 1 on the Exit Confirmation panel	126
5-19	Enter the refresh command on the Table Spaces panel	126
5-20	Refreshed TPAADMR2 shows type G	127
5-21	Enter the interpret line command next to TPAADMR2	127
5-22	Detailed information about the converted TPAADMR2 table space	128
5-23	Enter option 1 on the ADB2 panel	129
5-24	Locate the TPAADMR2 table space	130
5-25	Normal table space TPAADMR2	130
5-26	Enter T next to the TPAADMR2 table space	131
5-27	Issue LKEY line command against table TBADD304	131
5-28	Partitioning keys of the TBADD304 table displayed	132
5-29	Partition key column of the TBADD304 table	132
5-30	Enter the ALT command next to TPAADMR2	133
5-31	Enter the MAKEPBR command on ADB21SAR	133
5-32	Ready to convert to a partition-by-range UTS	134
5-33	Continue the PBR UTS conversion process	135
5-34	The TPAADMR2 table space will be modified	135
5-35	Specifying options for generating jobs for MAKEPBR	136
5-36	Specifying the data set name for the apply job	137
5-37	Generated analyze job of MAKEPBR	138
5-38	Select 1 on the Exit Confirmation	139
5-39	Enter refresh on ADB21S	139
5-40	TPAADMR2 changed to range-partitioned UTS	140
5-41	Enter the Interpret line command against TPAADMR2	140
5-42	Detailed information about the converted TPAADMR2	141
5-43	Enter T next to TPAADMR2 on ADB21S	141
5-44	Enter LKEY line command next to TBADD304	142
5-45	Partitioning key values for the TBADD304 table	142
6-1	List of CLONE tables	145
6-2	Find the related BASE table	146
6-3	BASE table display	146

6-4	List of BASE and CLONE tables	147
6-5	List tables related to the table space	147
6-6	List of tables: Result of issuing the T line command	148
6-7	ALL T command	148
6-8	Results of ALL T command	149
6-9	DS command display	149
6-10	AL - ADD CLONE	150
6-11	ADD CLONE	151
6-12	ALTER ... ADD CLONE syntax	151
6-13	ALTER stmt executed message	152
6-14	AL line command on BASE table	152
6-15	Select DROP CLONE	153
6-16	DROP CLONE	153
6-17	DROP CLONE Table	154
6-18	DROP CLONE	154
6-19	ALTER TABLE DROP CLONE syntax	155
6-20	Exchange data: XCHG line command	155
6-21	Exchange Data Between Base and Clone Tables	156
6-22	EXCHANGE DATA SQL syntax	156
6-23	EXCHANGE completed	157
6-24	Data set names for the BASE and CLONE tables	157
6-25	ICS IC Status	158
6-26	Interpret line command	158
6-27	SYSIBM.SYSCOPY details	159
6-28	Review /change options	160
6-29	COPY CLONE	161
6-30	DIS DB	162
7-1	Sample ALC error message	167
7-2	ALTER Analysis Options	168
7-3	Object Altering Report	169
7-4	Object Altering Report (continued)	170
7-5	Execute ALTER using SQL or through batch	170
7-6	ALT Job Flow	171
7-7	ALTER Build Analyze and Apply Job	172
7-8	ALT Job Flow (continued)	173
7-9	ALT Job Flow (continued)	174
7-10	ALT Job Flow (continued)	175
7-11	Alter Tables: Adding a table using wildcards	176
7-12	Select tables using the plus sign	177
7-13	Alter Tables panel with newly added objects	178
7-14	Alter a table using the A command	179
7-15	Alter tables using the REL command	180
7-16	Alter related objects	181
7-17	Alter Related: Alter a table space	181
7-18	Redefine a table space	182
7-19	Modified Alter Related panel	183
7-20	Modified Alter Tables Panel	183
7-21	Build Analyze and Apply Jobs	184
7-22	ALT line command example	187
7-23	Insert a new column named CELLPHONE	188
7-24	REL command to display related objects	189
7-25	Example of issuing the Alter command	190
7-26	Alter Related - View Column List	190

7-27	Add the columns to the view	191
7-28	Modification pending	191
7-29	List of changes	192
7-30	Alter Tables with schema and table name	193
7-31	List of tables matching the wildcard criteria	194
7-32	Alter Tables with additional tables	195
7-33	Alter Tables.	196
7-34	ALTER Table	197
7-35	Alter Related Objects	198
7-36	Alter Tables: Message regarding a CHG Rqd	198
7-37	Alter Tables.	199
7-38	Alter Tables.	200
7-39	Alter the table space	201
7-40	Redefine Table Space panel.	201
7-41	Alter Related with table space modification noted	202
7-42	Alter Related	202
7-43	DB2 10 for z/OS INCLUDE Column Feature	203
7-44	DB2 10 for z/OS INCLUDE Column Result.	204
7-45	Redefine Index - Space panel.	204
7-46	INDEX DDL.	205
7-47	Alter Related	206
7-48	Alter Tables.	206
7-49	Edit the view	207
7-50	Updated Alter Tables panel	208
7-51	Build Analyze and Apply Job panel.	209
7-52	ALTER Build Analyze and Apply Job (continued).	210
8-1	Specifying mask definitions in a data set	216
8-2	Mask options fields	217
8-3	Mask overwrite fields	217
8-4	Specifying mask definitions in DB2.	218
8-5	Entering the mask name	219
8-6	Initial mask detail entry panel	219
8-7	Mask detail lines	220
8-8	Mask lists	220
8-9	Managing a mask using Change Management	221
8-10	Selecting a mask.	221
8-11	Migration process flow	222
8-12	Database panel.	223
8-13	Objects for migration.	223
8-14	Adding another object.	224
8-15	Adding a database	224
8-16	Migration Parameters panel	225
8-17	Mask example.	226
8-18	Generated combined Migrate jobs	226
8-19	Generated single migrate jobs	227
8-20	Implicit database names	229
8-21	Initial Work Statement List panel	230
8-22	WSL library members	230
8-23	WSL cloning using masking	231
8-24	WSL Library after cloning	231
8-25	Interpreting the WSL.	232
8-26	Sample WSL Interpret output	233
9-1	DB2 Object Comparison Tool architecture	237

9-2 DB2 Administration Tool main menu	239
9-3 Change Management panel	240
9-4 Manage Versions panel	240
9-5 Insert Version Scope skeleton	240
9-6 Insert Version Scope completed	241
9-7 CM Manage Versions	241
9-8 CM Version Scope objects	241
9-9 Adding objects to the version scope	242
9-10 Indication that the Version Scope has been updated	242
9-11 Two version scopes defined	242
9-12 DB2 Administration Tool main menu	243
9-13 DB2 Object Comparison Tool Menu	243
9-14 Specify Compare Source	244
9-15 Specify Source Version Scope	244
9-16 CM Version Scope list.	245
9-17 DB2 Object Comparison Tool Menu	245
9-18 DB2 Object Comparison Tool Menu	246
9-19 Specify Compare Target	246
9-20 Specify Target Version Scopes	247
9-21 CM Version Scopes	247
9-22 DB2 Object Comparison Tool Menu: Specify Compare Masks	248
9-23 Specify Compare Masks	248
9-24 Insert Mask	248
9-25 Insert Mask	249
9-26 CM Mask Lines	249
9-27 CM Mask Lines	250
9-28 DB2 Object Comparison Tool Menu	250
9-29 DB2 Object Comparison Tool Menu: Generate compare job	251
9-30 Compare JCL that performs the Compare and produces the report.	254
9-31 DB2 Object Comparison Compare report	254
9-32 Generate Compare Jobs parameter panel	255
9-33 Example of the contents in a change file	256
10-1 Example of creating an WSL using the ALTER function.	260
10-2 Statement execution prompt panel	261
10-3 Manage Work Statement Lists panel	262
10-4 Work Statement List Library contents panel	263
10-5 Show Work Statement List panel	264
10-6 Insert work statement panel	265
10-7 Edit Work Statement List panel.	265
10-8 Clone Work Statement List panel	266
10-9 Interpret Work Statement List Options panel	267
10-10 Interpret Work Statement List panel	268
10-11 Confirm Checkpoint Record Removal panel.	269
10-12 HPU Unload Prompt panel	269
10-13 Specify Job Parameters panel	269
10-14 Specify Restart Information panel.	270
10-15 Edit WSL Restart Variable Overrides panel	271
10-16 Work Statement List Execution Output	273
10-17 Use the WSL primary command	275
10-18 Choose option 1 to show a WSL library	276
10-19 Run the BASEPRCA work statement list in batch	276
10-20 Specify N to keep the existing checkpoint records	277
10-21 Use DB2 Unload by answering N	277

10-22 Specify Job Parameters example	277
10-23 Example of Specify Restart Information panel	278
10-24 Alternate Specify Job Parameters example	278
10-25 Alternate Specify Restart Information panel	279
11-1 Displaying a locations database	282
11-2 Connecting to a remote subsystem	282
11-3 Connected to a remote subsystem	283
11-4 Local connection menu	284
11-5 ALL X command	285
11-6 ALL X command result	286
11-7 Issuing the REP command	287
11-8 Selecting the REP options	287
11-9 Session status.	290
11-10 Simple searching	291
11-11 Entering SEARCH conditions	292
11-12 Search results	293
11-13 DB2 command	293
11-14 DB2 command result	294
11-15 Execute SQL Statements panel	295
11-16 SQL entry panel	295
11-17 Using the hidden field	296
11-18 SQL results in DB2 Administration Tool panel	297
11-19 Table space utility generation	298
11-20 Other selection criteria	299
11-21 Finding the index.	300
11-22 Result of using other selection criteria to find an index.	300
11-23 Which column name to use.	301
12-1 Create/Drop/Label/Comment On Objects panel	307
12-2 Create CUST_COVERAGE table	308
12-3 Specifying table columns	309
12-4 Enter the first four columns and the SYS_STA column for CUST_COVERAGE	311
12-5 SYS_STA length is changed from 0 to 13	312
12-6 Issue the U line command next to the SYS_STA column.	313
12-7 Enter NO for WITH TIME ZONE	314
12-8 Operation type of SYS_STA changed to UPDATE.	315
12-9 Specify SYS_END column	316
12-10 Updating SYS_END column to WITHOUT TIMEZONE	317
12-11 Operation type of SYS_END column changed to UPDATE	318
12-12 Specifying BUS_STA, BUS_END, and the CREATE_ID column	319
12-13 Update CREATE_ID to WITHOUT TIMEZONE	320
12-14 Operation type of CREATE_ID is changed to UPDATE	321
12-15 TBLOPTS command.	322
12-16 Specify YES for both System period and Business period	323
12-17 Specify the starting and ending column from multiple columns	324
12-18 Period Column updated to reflect the B_1 and B_2 columns	324
12-19 Respecifying the ending time column	325
12-20 Error message displayed on the ADBP6CTB panel	325
12-21 End Period Column state was cleared from B_2 column	326
12-22 Enter the E line command next to the O_1 column	326
12-23 Period Column of O_1 changed to End	327
12-24 Select BUSINESS TIME Period Columns Row.	327
12-25 Specifying the Starting and Ending time column	328
12-26 Period Column updated for the starting and ending time column	328

12-27	Issue the Create command to create a table	329
12-28	TABLE CUST_COVERAGE created	330
12-29	Back to the DB2 Administration Menu	331
12-30	Issue TT command to locate the ADMR2.CUST_COVERAGE table	332
12-31	Issue DDL line command against CUST_COVERAGE table	332
12-32	Generated DDL for CUST_COVERAGE	333
12-33	DDL of CUSTOMER_COVERAGE table	334
12-34	Five rows of data in CUSTOMER_COVERAGE table	334
12-35	Issue AL line command.	335
12-36	Select the ADD column option	336
12-37	ADD system begin time column	337
12-38	ADD system end time column.	338
12-39	ADD transaction start ID column.	339
12-40	ADD PERIOD option.	340
12-41	ADD PERIOD SYSTEM_Time	340
12-42	Locate CUSTOMER_COVERAGE table	342
12-43	Issue the DDL line command to the CUSTOMER_COVERAGE table	343
12-44	Generated DDL of CUSTOMER_COVERAGE table	344
12-45	Modifying the DDL of the CUSTOMER_COVERAGE table	345
12-46	Execute the modified DDL	346
12-47	History table with table type T	346
12-48	Select ADD VERSION option on ADB21TA	347
12-49	Use look up to locate the history table	348
12-50	Select the history table from a list of tables	348
12-51	Specify the history table name	348
12-52	Locating the HS_CUSTOMER_COVERAGE table.	349
12-53	History table with a table type of H	350
12-54	Issue the L line command to show data rows	351
12-55	CUSTOMER_COVERAGE system period temporal base table	351
12-56	HS_CUSTOMER_COVERAGE table contains archived rows	352
12-57	Enhanced BASE line command	352
12-58	Base temporal table	353
12-59	ALT command not allowed for history table	353
12-60	DROP VERSIONING option	354
12-61	HS_CUSTOMER_COVERAGE with a table type of T	355
13-1	DB2 system authorities	358
13-2	Updating DSNZPARM SECADM user ID	360
13-3	Authorization options	361
13-4	System Privileges Authorizations panel	362
13-5	Grant System Privileges panel	363
13-6	Built GRANT statement.	364
13-7	Updated system authorizations.	364
13-8	Revoking system authorities	365
13-9	Revoke Impact Report panel	365
13-10	Revoke table authorities	367
13-11	Revoke dependent authorities	368
13-12	Selecting table authorizations	369
13-13	Table authorizations for DSN8A10	370
13-14	Entering grantees	370
13-15	Statement Execution Prompt	371
13-16	Create/Drop/Label/Comment on Objects panel	373
13-17	Initial Create Column Mask panel	374
13-18	Create Column Mask panel	375

13-19	Create Column Mask panel with RETURN (Expression)	376
13-20	Statement Execution Prompt panel	377
13-21	Altering a Trigger to SECURE	378
13-22	Activate column access control	379
13-23	Activating a mask from the column list	380
13-24	Object authorization options	380
13-25	Column Mask panel	381
13-26	Mask definition	381
13-27	Column mask dependent objects	382
13-28	Interpret column masks	382
13-29	Showing permissions on a table list	383
13-30	Dropping column mask	384
13-31	Initial Create Row Permission panel	385
13-32	Create Row Permission panel	386
13-33	Create Row Permission panel with search condition	387
13-34	Executing the row permission	388
13-35	Activate row access control	389
13-36	Activating row permissions from Row Permissions panel	390
13-37	Row permissions from a table list	390
13-38	Row permissions list	391
13-39	Generate a row permission	392
13-40	Row permission Environmental Variables panel	392
13-41	Interpreting the ENV variables	393
13-42	Row permissions after deactivation	394
13-43	Drop Row Permission panel	394
13-44	Trusted Contexts panel	395
13-45	Creating a trusted context: Part 1	396
13-46	Creating a trusted context: Part 2	397
13-47	Creating a trusted context: Part 3	397
13-48	Create a local trusted context: Part 1	398
13-49	Create a local trusted context: Part 2	399
13-50	Adding users to a trusted context	400
13-51	Adding additional users to the trusted context	401
13-52	Adding Trusted Context Attribute	402
13-53	Altering or adding attributes to a trusted context	402
13-54	Dropping an attribute from a trusted context	403
13-55	Trusted Context IDs	403
13-56	Altering attributes of an ID	404
13-57	Trusted context list	405
13-58	Create/Drop/Label/Comment On Objects panel	406
13-59	Creating a role	407
13-60	Viewing roles	407
13-61	From role to trusted context	408
13-62	Dropping a role	409
13-63	Audit Policy list	410
13-64	Inserting an audit policy	411
13-65	Define database auditing policy	414
13-66	Auditing changes to the audit tables	415
13-67	Interpret Policy 2	416
14-1	AL line command: Unique index	418
14-2	ADDCOL	419
14-3	Add as INCLUDE	419
14-4	ALTER statement of INCLUDE column	420

14-5 Column included: Results in a +610	420
14-6 Create index: Utilities	421
14-7 Adding an INCLUDE column using the ALT line command	422
14-8 INCLUDE column	423
14-9 INCLUDE column using ALT	424
14-10 ALT INCLUDE column	425
14-11 Clustering Index Confirmation panel.	425
14-12 Redefine Index	426
14-13 Index DDL	426
14-14 Alter Tables panel	427
14-15 Build Analyze and Apply panel	428
14-16 Data set Existence Check panel	428
14-17 Work Statement List library	429
14-18 WSL batch execution 610 warning message	430
14-19 Online WSL execution output	431
14-20 Removing an INCLUDE column	432
14-21 Generated SQL syntax for the index.	433
14-22 Creating a LOB column.	434
14-23 Creating an inline LOB	435
14-24 Updating inline length	436
14-25 Inline LOB size	437
14-26 Selecting a table to alter	438
14-27 Selecting a column to alter	438
14-28 Altering inline length	439
14-29 Column update marker	440
14-30 ALTER Analysis Options panel	440
14-31 Build Analyze and Apply Job panel.	441
14-32 Creating an XML table	443
14-33 Column details	444
14-34 Adding XML to a table.	445
14-35 Adding XML to a table: AL	445
14-36 XML column details.	446
14-37 Adding XMLSCHEMA validation.	447
14-38 Creating an Index on an XML column.	448
14-39 Naming an index	448
14-40 Create Index panel	449
14-41 Selecting index column	450
14-42 Entering XMLPATTERN	450
14-43 XMLPATTERN within an index	451
14-44 Generating the XMLPATTERN index	451
14-45 System Administration panel	453
14-46 Change DB2 System Checkpoint Frequency panel	453
14-47 SET LOG NEWLOG COPY	454
14-48 ALTER TABLE ... ADD PART syntax	454
14-49 Error message when exceeding the number of MAXPARTITIONS	455
14-50 TIMESTAMP: YYYY-MM-DD-HH.MN.SS.MICROSEC.	456
14-51 TIMESTAMP WITH TIME ZONE	456
14-52 AL ADD column	457
14-53 Enter the Column name, Column type, Scale, and WITH TIME ZONE YES	458
14-54 ALTER statement	459
14-55 SQL statement generated to add a TIMESTAMP WITH TIME ZONE column using ALT.	459
14-56 TIMESTAMP WITH TIME ZONE Column Type TIMESTZ	460

14-57	CONCURRENTACCESSRES	461
14-58	DB2 Administration Tool Menu panel	462
14-59	Alter buffer pools	463
14-60	Alter Buffer Pools	464
14-61	Alter Buffer Pool	464
15-1	System Catalog PDC command	467
15-2	DB2 Pending Definition Changes panel	467
15-3	Drop DB2 Pending Definition Changes	468
15-4	ALTER TABLESPACE with DROP PENDING CHANGES	468
15-5	Interpret line command	469
15-6	SQ line command	469
15-7	Original definition of the table space	470
15-8	Modified table space definition	471
15-9	Statement Execution Prompt panel	471
15-10	AL Table Space with +610 message	472
15-11	DIS command	473
15-12	Results of the DIS command	474
15-13	Table Spaces	475
15-14	DDL for Table Space	476
15-15	DDL for Table Space (continued)	476
15-16	System Catalog panel	477
15-17	DB2 Pending Definition Changes panel	477
15-18	AL line command next to a table space	478
15-19	Original definition	479
15-20	Updated fields	479
15-21	ALTER statement generated	480
15-22	Negative return code	480
15-23	Object in pending state	481
15-24	SYSIBM.SYSPENDINGDDL table	481
16-1	GEN parameter panel	485
16-2	SQL Execute & Copy panel	485
16-3	GEN Extracted DDL in a queue	486
16-4	GEN parameter panel	487
16-5	GEN output: Include DB2 Pending Changes = Yes	487
16-6	GEN output: Include DB2 Pending Changes = Alter	488
16-7	GEN output: Include DB2 Pending Changes = ONLY	488
16-8	GEN SQL Statement Options	489
16-9	System Catalog panel	490
16-10	Authorization options	491
16-11	Packages QUALIFIER	492
16-12	Packages PLANMGMT	493
16-13	Performance Queries panel	494
16-14	List of Indexes: Unused	495

Tables

2-1 Underlying components of DB2 Administration Tool	14
2-2 DB2 Administration Tool product libraries	14
2-3 DB2 Object Compare product libraries	15
2-4 Checkpoint database table uses	15
2-5 Change management database table uses	16
2-6 Catalog copy version table uses	17
2-7 DB2 Administration Tool plan names	17
2-8 DB2 Administration Tool collection names	18
2-9 Example environments	34
2-10 Planning worksheet section pertaining to product libraries	35
2-11 Checkpoint database-related jobs in SADBSAMP for a new installation	37
2-12 Checkpoint database object names	37
2-13 Planning worksheet section pertaining to the checkpoint database	38
2-14 Segmented temporary table space-related jobs in SADBSAMP	40
2-15 Change management database-related jobs in SADBSAMP	40
2-16 Change management database object names	41
2-17 Planning worksheet section pertaining to the change management database	42
2-18 Bind-related jobs in SADBSAMP	44
2-19 Planning worksheet section pertaining to the binds of product plans and packages	44
2-20 Variable block-related jobs in SADBSAMP	46
2-21 Planning worksheet section pertaining to the creation of the variable-block data sets	46
2-22 Planning worksheet section pertaining to the invocation of the ADBL CLIST	47
2-23 Planning worksheet section pertaining to customizing the operating environment on ISPF	49
2-24 Catalog copy-related jobs in SADBSAMP	58
2-25 Planning worksheet section pertaining to the catalog copy version database	58
2-26 RUNSTATS views-related jobs in SADBSAMP	59
2-27 Planning worksheet section pertaining to the creating RUNSTATS update views	59
2-28 Grant-related jobs in SADBSAMP	60
2-29 Planning worksheet section pertaining to the granting of SELECT authority on the DB2 catalog	60
2-30 SDB2 catalog index-related jobs in SADBSAMP	61
2-31 Planning worksheet section pertaining to the creation of additional indexes on the DB2 catalog	61
2-32 Distributed support-related jobs in SADBSAMP	62
2-33 Planning worksheet section pertaining to enabling DB2 Administration Tool distributed support	62
7-1 ALT command replacements	166
7-2 ALTER Analysis Options panel parameters	168
7-3 Build, Analyze, and Apply Parameter panel options	184
8-1 Possible job names	227
9-1 Generate Compare Jobs parameters	251
11-1 Valid ALL objects	286
13-1 Object code	366
13-2 Jobname attributes	399
13-3 Audit policy fields	411
A-1 DB2 Administration Tool install and upgrade planning worksheet	500
A-2 Translation mask names	504

A-3 Migration parameter details	506
---------------------------------------	-----

Examples

2-1 Query to find temporary databases	22
2-2 Query to find segmented temporary table spaces	22
2-3 Initial example of the DB2 subsystem customization edit session	52
2-4 Sections for the common and subsystem-specific parameters	53
3-1 Customization ISPF table edit	68
3-2 DB2 Administration Tool defaults	71
7-1 Output report	211
7-2 Output report (continued)	212
8-1 Ordering mask fields	218
8-2 Create Report from Migrate example	228
11-1 Report output	288
11-2 Index data set names	299
13-1 Generated Revoke Dependent statement	368
13-2 Generate Revoke Not Dependent statement	368
13-3 Copying authorizations batch job	371
13-4 Entering Column Mask Expressions	375
13-5 Generated Create Mask statement	377
13-6 Generated Column Access statement	379
13-7 Deactivate a column mask	383
13-8 Drop column mask statement	384
13-9 Entering row permission search conditions	386
13-10 Generated row permission	388
13-11 Generated row access statement	389
13-12 Deactivate row permissions	393
13-13 Generated drop row permission	394
13-14 Generated trusted context	398
13-15 Generated local trusted context	399
13-16 Generated local trusted context with users	401
13-17 Drop attribute statement	403
13-18 Generated Replace command	404
13-19 Drop ID command	404
13-20 Enable a trusted context	405
13-21 Generate trusted context used by a role	408
13-22 Starting trace with an audit policy	414
14-1 Generated WSL	441
14-2 Generated alter table	447
14-3 XML Modifier in Create Table Statement	447
14-4 XMLPATTERN Index DDL	452

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®

DB2®

DRDA®

IBM®

Informix®

MVS™

OS/390®

pureXML®


QMFTM

Query Management Facility™

RACF®

Redbooks®

Redpaper™

Redbooks (logo) ®

System z®

System/390®

z/OS®

The following terms are trademarks of other companies:

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Today's business environment has increased in the complexity and rate of change that a database administrator must control. The ability to respond quickly to a changing environment is constantly challenged by the explosion of data growth combined with a decline in an experienced work staff.

The IBM® DB2® Administration Tool for z/OS® Version 10 helps you become productive from Day 1 with DB2 10 for z/OS by using performance savings right away, lowering the CPU costs while reducing the batch window. Users experience higher data availability by easily managing online schema changes, including additional columns to indexes to use index-only access.

Customers are able to experience higher data availability through simplified recovery operations:

- ▶ Access new functionality in DB2 10 for z/OS to lower costs and improve efficiency both before, during, and after the DB2 migration process.
- ▶ Maximize the performance of your key DB2 business applications to speed their deployment in DB2 10 for z/OS.
- ▶ Improve the productivity and efficiency of your staff when DB2 10 for z/OS is running.

This IBM Redbooks® publication highlights the data administration enhancements introduced by DB2 Administration Tool for z/OS Version 10 by providing scenarios of their use with the new functions provided by DB2 10 for z/OS.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Paolo Bruni is a DB2 Information Management Project Leader at the International Technical Support Organization based at the Silicon Valley Lab. He has authored several IBM Redbooks publications about DB2 for z/OS and related tools and has conducted workshops and seminars worldwide. During Paolo's many years with IBM, in development and in the field, his work has been mostly related to database systems.

Tom Crocker is a Senior IT Specialist working in the United Kingdom as a Client Technical Advisor specializing in DB2 Tools and DB2 Engine. Tom has over 20 years experience in data management and has worked for IBM for the last 10 years, first as a DB2 DBA supporting SAP on OS/390® and for the last 8 years in Technical Pre-Sales. Before joining IBM, he worked as an independent consultant and has experience in the insurance, banking, and retail industries. He has previously authored an IBM Redbooks publication on DB2 7 utilities.

Elaine Morelli is an Executive IT Specialist with the North America IBM System z® Database Tools Team. Elaine's expertise is in the DB2 database tools arena with a specialization in DB2 administration. Elaine has over 25 years IT experience, 11 of which have been at IBM in a technical sales specialist role. In the past, she has been an Application Developer, a DOS/VSE Systems Programmer, a DBA for DB2, IDMS, Informix®, and Oracle. Elaine has presented at various local user groups and technical conferences across the United States. She is the IBM sponsor for the Tampa Bay Relational User Group.

Richard Schaufuss is a Software Engineer working as a developer on the DB2 Administration Tool for z/OS. Rich joined IBM 11 years ago and, after 4 years of working on DB2 for z/OS development, has moved to DB2 tools for z/OS, first supporting customers in the field and recently as developer. He has recently participated in designing and writing the certification exams for DB2 10 for z/OS.

Jane Yang is a Quality Assurance Architect and technical lead for the DB2 Administration Tool for z/OS. She has 6 years of experience in this role. Prior to joining the QA team, Jane worked for 13 years in the DB2 for z/OS performance department. She has published numerous technical reports and taught customer classes in the area of DRDA® distributed database processing and stored procedures.

Thanks to the following people for their contributions to this project:

Rich Conway
Bob Haimowitz
Emma Jacobs

International Technical Support Organization

Melissa Biggs
Jay Bruce
Chris Burgman
Jim Kennelly
June Kin
Joseph Reynolds
IBM Silicon Valley Lab

Tom Ulveman Jensen
IBM Denmark

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>



Part 1

Introduction

In this part, we introduce the DB2 administration tools. This part consists of the following chapter: Chapter 1, “DB2 administration tools at a glance” on page 3.



DB2 administration tools at a glance

The primary tools provided by IBM for DBAs working with DB2 for z/OS environments are the IBM DB2 Administration Tool for z/OS and IBM DB2 Object Comparison Tool for z/OS.

We do not give a detailed description of every function that is part of both products. This would be a challenging task, especially with the DB2 Administration Tool for z/OS. For detailed information about all the functions the tools provide, refer to the following books:

- ▶ *DB2 Administration Tool for z/OS User's Guide and Reference Version 10 Release 1*, SC19-3033
- ▶ *DB2 Object Comparison Tool for z/OS User's Guide Version 10 Release 1*, SC19-3037

In this book, we focus on the functions provided by the tools that the database administrators use in their daily job when managing DB2 objects, such as creating, changing, and migrating. In particular, we highlight some best practices and the use of the tools when implementing the functions made available with DB2 10 for z/OS.

This chapter is meant to be a quick overview of the tools and contains the following topics:

- ▶ DB2 Administration Tool for z/OS
- ▶ DB2 Object Comparison Tool for z/OS
- ▶ Administration evolution and support for DB2 10 for z/OS

1.1 DB2 Administration Tool for z/OS

DB2 Administration Tool for z/OS, or DB2 Administration Tool, as we refer to it throughout this book, provides a comprehensive set of functions to help you manage your DB2 efficiently and effectively, even though your IT environment may be complex. It offers a comprehensive set of database administration functions and manages changes to DB2 objects.

DB2 Administration Tool provides in-depth catalog navigation by displaying and interpreting objects in the DB2 catalog and executing dynamic SQL statements. It is integrated with other DB2 tools (such as SQL Performance Analyzer, High Performance Unload, and Table Editor) to simplify the creation of DB2 utility jobs. This integration creates additional functionality with product-specific line commands for table editing, SQL cost analysis, and path check analysis.

Figure 1-1 describes the major DB2 Administration Tool functions.

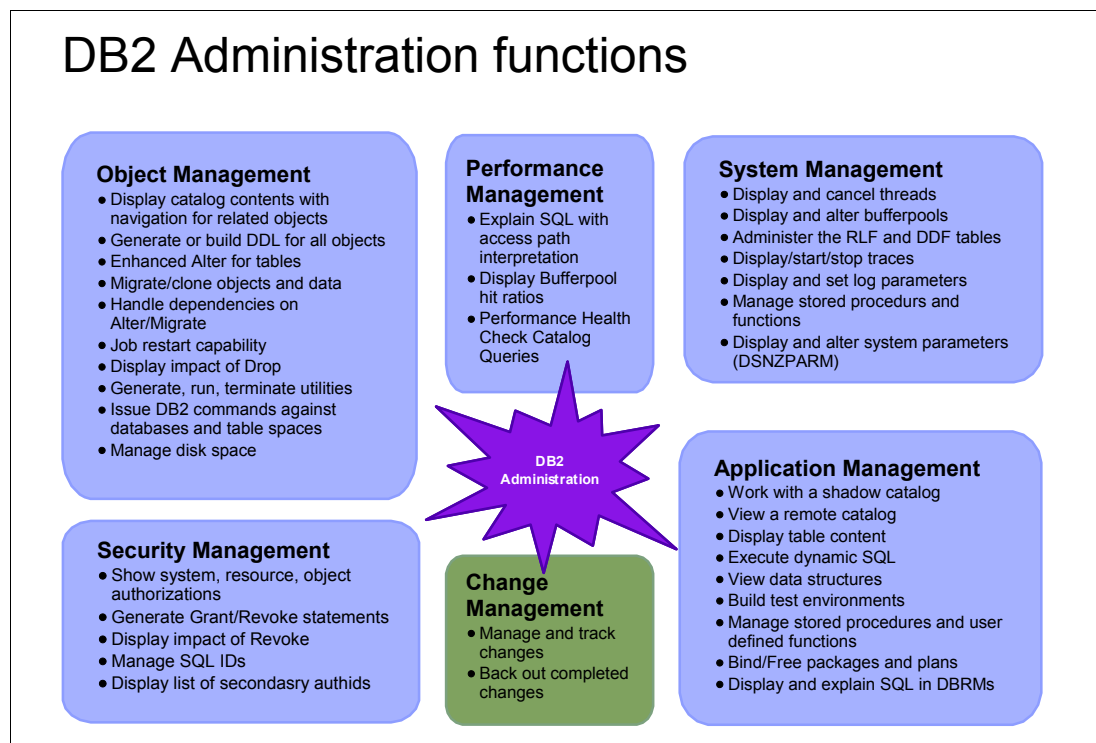


Figure 1-1 DB2 Administration Tool overview

From the DB2 Administration Tool panels, you can issue primary commands and line commands. Primary commands can be issued from the Command line on the panel and their syntax is explained in the help panels. Line commands are issued from ISPF table display panels and specify an operation that is performed on the information that is displayed on the row. You also can define your own line commands during installation.

Object management

Because DB2 Administration Tool provides in-depth DB2 catalog navigation, it minimizes the time required to review the catalog. Objects in the catalog are displayed and interpreted, and relevant catalog information is presented logically. You can issue any DB2 command, including BIND, REBIND, and FREE selected plans and packages.

DB2 Administration Tool is integrated with DB2 utilities to simplify the creation of DB2 utility jobs. Job control language (JCL) can be generated for DB2 utilities to run them in batch. Utility statements can also be included in a work statement list (WSL) to be run at another time or on another subsystem. Using LISTDEFS and TEMPLATES is also supported.

The DB2 Administration Tool ALTER and MIGRATE functions can simplify administration tasks. After using the ALTER function to specify desired changes, the tool generates the jobs required to implement these changes. These jobs unload the data, recreate the table, and reload the data if needed. The tool handles all object dependencies during ALTER and MIGRATE. And, after the MIGRATE function has defined the target DB2 subsystem, DB2 Administration Tool creates the jobs required to copy definitions and data to the target:

- ▶ The ALTER function lets you change the name and attributes of a table or column, insert new columns and drop existing columns, as well as altering table spaces, indexes, views, foreign keys, and databases.
- ▶ The MIGRATE function facilitates the copying of all the objects and data in one or more databases or table spaces to another DB2 subsystem.

Prompt options can be activated for five types of statements: definition, authorize and update SQL, DB2 commands, and DSN commands. These options enable you to edit or execute the statements, put them in work statements, or run them in batch jobs.

An ALTER of primary key characteristics can be propagated to foreign keys.

DROP impact analysis is provided to prevent inadvertent data loss when dropping tables.

DB2 Administration Tool also helps you manage disk space by gathering Virtual Storage Access Method (VSAM) related information in a convenient way, and also helps you with space estimation of your DB2 objects.

Security management

DB2 Administration Tool allows you to display authorizations that have been granted on any type of DB2 object, and enables you to REVOKE these authorizations or GRANT new ones.

The tool also provides REVOKE impact analysis to prevent inadvertent loss of authorizations when revoking authorities.

It allows you to display the list of secondary authorization IDs, and to change your SQLID.

Performance management

DB2 Administration Tool has a built-in EXPLAIN function that allows you to EXPLAIN a query and provides an interpretation of the PLAN_TABLE output into English sentences.

The tool also comes with a set of performance health check catalog queries.

Change management

The Change Management (CM) function can be used to manage and track the changes to the DB2 objects. It allows groups of users to collaborate in building changes by managing information through a series of DB2 tables stored in a single database. Making changes through the DB2 Administration Tool Change Management process provides a convenient audit trail where the changes are in the process, and you can understand which changes have been deployed.

Completed changes that have been made through Change Management can be rolled back (one at a time).

System management

DB2 Administration Tool allows you to display and cancel threads, display and alter buffer pools, and display, start, and stop DB2 traces, and set and display the logs.

It also provides a convenient way to administer RLF and DDF tables.

DB2 Administration Tool manages stored procedure operations, such as creating, displaying, or altering, issues the DB2 START and STOP PROCEDURE command, and shows statistics for stored procedures that are accessed by DB2 applications. Other system administration functions such as managing functions are also supported.

Dynamic DSNZPARM changes enable you to display current DSNZPARMs, change parameters, generate new DSNZPARM modules with changes, and activate those changes in DB2.

Application management

DB2 Administration Tool allows you to work with a copy of the DB2 catalog to avoid contention and other performance problems on the actual catalog. With DB2 10 for z/OS, contention can be reduced by setting the USE CURRENTLY COMMITTED attribute on all dynamic SQL statements to YES.

You can access a remote DB2 catalog, as long as a DDF connection exists between the systems. This connection enables you to do centralized management of all your DB2 subsystems through a single DB2 Administration Tool session.

You can also execute any dynamic SQL statement through the DB2 Administration Tool, or you can invoke SPUFI from within the DB2 Administration Tool.

The integration of DB2 Administration Tool with other DB2 tools creates additional functionality with product-specific line commands for table editing, SQL cost analysis, and path check analysis. Through its DB2 tools launch pad, DB2 Administration Tool for z/OS is the central access point for the current tools with an interactive system productivity facility (ISPF) interface, such as DB2 Table Editor, DB2 SQL Performance Analyzer, and DB2 High Performance Unload.

1.2 DB2 Object Comparison Tool for z/OS

When you rely on IBM DB2 for z/OS as your production database for business-critical information, it makes sense to have a duplicate system for testing and development. Although the purpose of a development system is to make changes and test them and the DB2 definitions are designed to do so, the challenge is to propagate the changes up the hierarchy, through test and finally to production in a controlled way.

This is where IBM DB2 Object Comparison Tool for z/OS, called DB2 Object Comparison Tool in this book, can help you. With a rich set of functions, DB2 Object Comparison Tool makes it easy to compare existing objects (and dependent objects) from different sources and subsequently synchronize these sources by staging and propagating changes between the environments.

Figure 1-2 gives you an overview of the DB2 Object Comparison Tool functions.

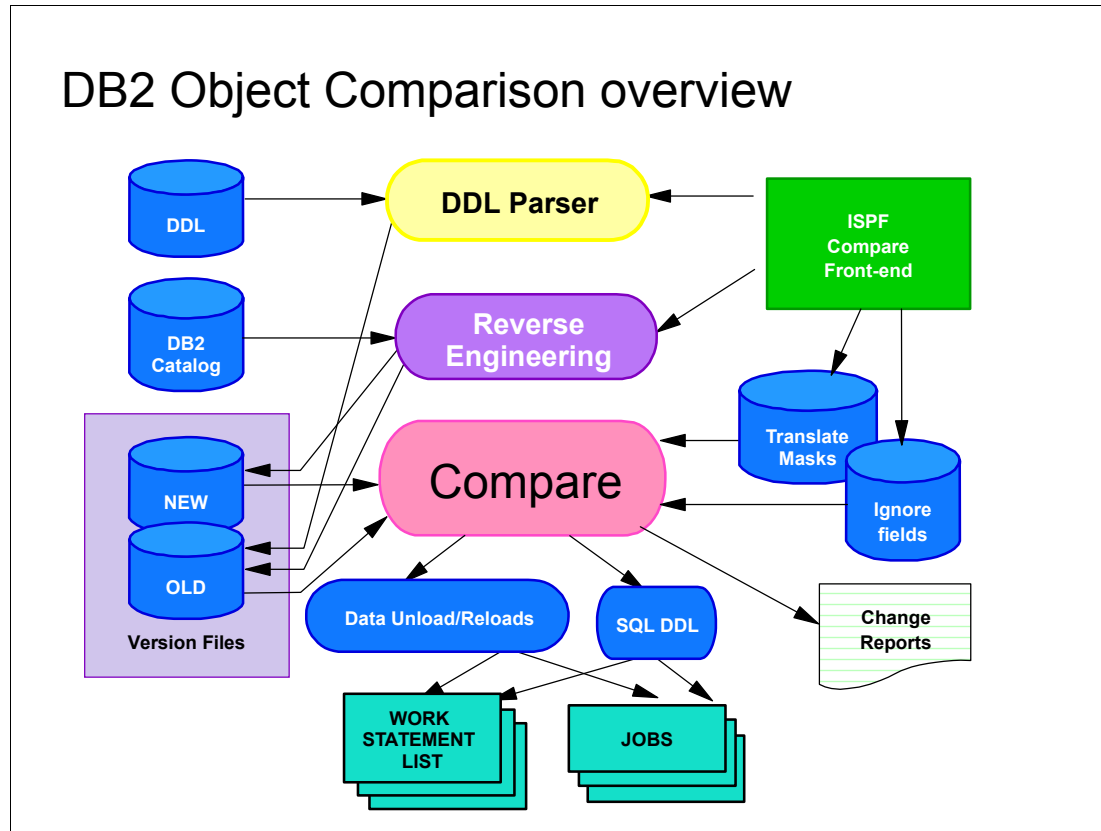


Figure 1-2 DB2 Object Comparison Tool overview

The DB2 Object Comparison Tool consists of:

- ▶ An interactive system productivity facility (ISPF) front end for specification of the objects to be compared.
- ▶ A DB2 catalog extract function, GEN, that pulls definitions from the catalog into a version file to support the comparison process. This function, in addition to extracting the DDL for objects, can also generate the DCL for all authorizations to the objects and the DML for the catalog statistics for the objects
- ▶ A DDL extract function that reads DDL statements and converts them into a version file.
- ▶ A batch compare function that compares two version files, produces several reports describing the differences found, and generates the information required to apply changes to the target
- ▶ A batch job generator that provides everything necessary to apply the changes to the target

DB2 Object Comparison Tool enables you to keep the object definitions in your DB2 test and production catalogs synchronized.

The tool's masking and ignore files, which address intentional differences or naming conventions, allow you to compare only the real differences that might exist between objects.

Using DB2 Object Comparison Tool, you can compare:

- ▶ Objects from a DB2 catalog to objects residing in another DB2 catalog or in the same catalog with different names
- ▶ A file with data definition language (DDL) to another file with DDL
- ▶ A file with DDL to objects in a DB2 catalog
- ▶ The DB2 catalog against a file

Whether you specify objects at the database level, the table space level, or the table level, all dependent objects, such as views and indexes, are included in the comparison. After you select an object, that object and its dependents are extracted from the catalog or DDL file and placed in a sequential data set, creating a *version* file (the source). You can then select the object to which the source should be compared. This object is extracted from its catalog and placed in a separate sequential data set creating another version file (the target).

A comparison of the source and target data sets results in the creation of a difference file. DB2 Object Comparison Tool generates a report of these differences for your review. Reports can be selected by the DBAs to choose the information included. Upon reviewing the report, you can direct the DDL generated for target changes to apply jobs. Apply jobs can be stored in a work statement list or a PDS file and then propagated to several remote sites. Changes in a test environment can be automatically migrated to the development or production environment.

DB2 Object Comparison Tool runs as an extension of DB2 Administration Tool. This enables you to take advantage of the functions in DB2 Administration Tool for z/OS.

DB2 Administration Tool, used in conjunction with DB2 Object Comparison Tool, offers a solution for handling complex processes associated with Change Management. How to work with the provided comprehensive Change Management processes is discussed in *DB2 for z/OS Administration Tools for Enhanced Change Management*, SG24-7441.

1.3 Administration evolution and support for DB2 10 for z/OS

IBM DB2 Administration Tool for z/OS enables the day-to-day tasks that are associated with managing a DB2 database:

- ▶ It simplifies the complex tasks associated with safely managing DB2 objects and schema throughout the application life cycle with the least possible impact to availability.
- ▶ It allows users to navigate the DB2 catalog quickly and easily.
- ▶ It builds and executes dynamic SQL statements without knowing the exact SQL syntax.
- ▶ It manages and tracks changes made to DB2 object definitions resolving any potential conflicts prior to execution.
- ▶ It helps build DB2 commands to execute against databases and tables.
- ▶ It enables users to create, alter, migrate, drop and reverse engineer DB2 objects.
- ▶ It builds and executes utility jobs, allowing users to take advantage of LISTDEFS and TEMPLATES for increased productivity.

Version 10.1 of DB2 Administration Tool focuses on support for DB2 10 for z/OS, including GEN and catalog navigation support, as well as support for many new DB2 10 for z/OS object types and attributes. Since the previous Version 7.2, the main product functions new in V10 include:

- ▶ ALT single interface to change
- ▶ Autonomic utilities
- ▶ NOPAD support for faster UNLOAD support extended to change
- ▶ Package versioning for Versions 8, 9, and 10

All the DB2 9 for z/OS and DB2 8 for z/OS functions are all supported.

For DB2 Administration Tool support of DB2 10 for z/OS functions shown in this book, we have used the general availability code with APARs PM27184 for DB2 Administration Tool and PM27186 for DB2 Object Comparison Tool.



Part 2

Installation and customization

In this part, we discuss the installation and tailoring of DB2 Administration Tool for z/OS to your environment.

This part contains the following chapters:

- ▶ Chapter 2, “Product setup” on page 13
- ▶ Chapter 3, “Product parameters” on page 65



Product setup

In this chapter, we provide information about installing DB2 Administration Tool and DB2 Object Compare. With extensive examples, we include a case of a new installation, a case of updating an existing installation of the two tools, and a case of enabling a new release of DB2 for access by both DB2 Administration Tool and DB2 Object Comparison Tool.

This chapter contains the following topics:

- ▶ The underlying components of DB2 Administration Tool
- ▶ Planning the installation of DB2 Administration Tool
- ▶ Installing a new image of DB2 Administration Tool

2.1 The underlying components of DB2 Administration Tool

The underlying components of DB2 Administration Tool are the product libraries containing the operational modules, various newly created database objects used by the batch restart and change management processes, and objects added to the DB2 catalog for operational and performance purposes.

The components are summarized in Table 2-1. We provide further information in the following sections.

Table 2-1 Underlying components of DB2 Administration Tool

Objects	What is it used for?
Product libraries	Contains the modules used in the operation of DB2 Administration Tool.
Checkpoint database	Used to manage the restart of the DB2 Batch Restart Programs (ADBTEP2 and ADBTEPX).
Change management database	Used to manage changes being done in the DB2 Administration Tool and DB2 Object Compare.
Segmented temporary table space	Used for the declared temporary objects that DB2 Administration Tool needs to allocate for various tasks. Only required for DB2 8 subsystems where you do not already have a segmented temporary table space with a buffer pool size of at least 8 KB.
DB2 catalog copy (optional)	Used to store copies of the DB2 catalog if you want to have the option to operate against a copy or you want to prevent operations against the real DB2 catalog.
Additional views against DB2 Catalog objects (optional)	Used to enable the updating of RUNSTATS information for user-objects.
Additional indexes on DB2 Catalog objects (optional)	Used to improve performance for DB2 Administration in its access of the DB2 catalog.
Product plans	Names of the plans provided in the distribution libraries.
Product collections	Names of the collections provided in the distribution libraries.

2.1.1 Product libraries

The product libraries contain the modules necessary for the overall operation of the product and must be accessible on disk from the DB2 subsystems/members/data sharing group onto which the tool is to be installed. These are the SMP/E target libraries created or modified dependant upon the naming convention defined by the SMP/E zone. There is one set of product libraries for DB2 Administration Tool and one for DB2 Object Compare, as shown in Table 2-2 and Table 2-3 on page 15, respectively.

Table 2-2 DB2 Administration Tool product libraries

Data set name	Data set contents
SADBBASE	SMP/E Job Control Language (JCL) Members
SADBCLST	TSO CLISTs
SADBDBRM	DB2 DBRMs
SADBEXEC	REXX EXEC Modules

Data set name	Data set contents
SADBLINK	Load modules that require APF-authorization
SADBLLIB	Other product load modules
SADBMLIB	ISPF Messages
SADBPLIB	ISPF Panels
SADBSAMP	JCL Members used to perform product set up tasks
SADBSLIB	ISPF File-Tailoring Skeletons
SADBTLIB	ISPF Tables

Table 2-3 DB2 Object Compare product libraries

Data Set Name	Data set contents
SGOCBASE	SMP/E Job Control Language Members
SGOCCLST	TSO CLISTs
SGOCEXEC	REXX EXEC Modules
SGOCLLIB	Product Load Modules
SGOCMLIB	ISPF Messages
SGOCPLIB	ISPF Panels
SGOCSAMP	JCL Members used to perform product set up tasks
SGOCSLIB	ISPF File-Tailoring Skeletons
SGOCTLIB	ISPF Tables

2.1.2 Checkpoint database

The checkpoint database is made of multiple objects that the batch restart programs, ADBTEP2 and ADBTEPX, require to resume execution if any of the statements executed in the input stream should fail. All of the objects that are part of the checkpoint database are put into a single DB2 database. The tables and their purposes are listed in Table 2-4.

Table 2-4 Checkpoint database table uses

Table name	Purpose
ADBCHKPT	Primary checkpoint record containing the environment variable setting and last attempted action information along with the restart number used by ADBTEP2/ADBTEPX.
ADBPART	Collects granular restart information pertaining to the partition level.
ADBCHK	Collects restart information pertaining to the CHECK utility.
ADBHOLD	Used to store restart information in regard to the failing command.
ADBDBETLIST	Used to store the contents of the declared global temporary table in regard to pending changes to objects in the event a restart needs to occur.

2.1.3 Change management database

The change management database is made of multiple objects that are used to store various objects used by DB2 Administration Tool change management processes and DB2 Object Compare (Table 2-5) and it is also used to store imported versions and can be used to store mask and ignore files.

Table 2-5 Change management database table uses

Table name	Purpose
ADBCHG	Primary records that identify changes.
ADBCHGS	Granular information defining changes by object.
ADBCHGSR	Granular information defining changes by object necessary to recover from a change.
ADBCPREREQ	Prerequisite change information.
ADBCMASK	Header record for a change mask.
ADBCMASKS	Granular information for a change mask.
ADBCIGNORE	Header record for one or more ignore options.
ADBCIGNORES	Granular information for each ignore option.
ADBCVERSION	Header record for versions stored in the database.
ADBCVERLINES	Granular information for the lines contained in a version.
ADBCVERSCOPE	Header record for a version scope.
ADBCVERSCOPES	Granular information defining the content of a version scope.
ADBCID	Change management IDs.

2.1.4 Segmented temporary table space

If you are using the DB2 Administration Tool with DB2 8 for z/OS, it is necessary to have a segmented table space in a temporary database with a buffer pool size of at least 8 KB. This object is used for the various declared temporary tables that DB2 Administration builds.

2.1.5 DB2 catalog copy version database (optional)

DB2 Administration Tool can be configured to use copies of the DB2 catalog rather than the live copy. This configuration can be helpful if you want to prevent users from accessing the live DB2 catalog, you want to maintain multiple copies of a DB2 catalog, or you want to store a local copy of the DB2 catalog of a remote subsystem.

DB2 Administration Tool manages these copies using a catalog copy version database that must be created if you want to use this functionality, otherwise, it is optional. See Table 2-6. These objects can also be created at a later time and are not necessary for the general operation of the product.

Table 2-6 Catalog copy version table uses

Table name	Purpose
ADBCATVT	Catalog copy information

2.1.6 Additional views against DB2 catalog objects (optional)

The functionality exists in DB2 Administration Tool to allow users to manually and optionally update RUNSTATS information beyond what can normally be updated in DB2 itself. This is managed by creating views against objects in the DB2 catalog that allow the object creator to update these RUNSTATS. If you just want to update RUNSTATS in the traditional sense (that is, through the RUNSTATS utility or real-time statistics), these objects do not need to be created,

2.1.7 Additional indexes on DB2 catalog objects (optional)

The queries that DB2 Administration Tool uses in some cases do not perform well because there are indexes missing from the delivered DB2 catalog. Depending on the release level of DB2, we suggest that certain indexes be created on the DB2 catalog to make the DB2 Administration Tool queries perform more efficiently. While this is an optional step, it is highly recommended,

2.1.8 Product plans

Several plans are provided with the DB2 Administration Tool. Their names and purposes are shown in Table 2-7.

Table 2-7 DB2 Administration Tool plan names

Plan	Purpose
ADB	Main DB2 Administration Tool plan.
ADB2GEN	Object data definition language (DDL) reverse engineering.
ADB27AC	Loads control card and data conversion.
ADB2CID	Changes management promotion.
ADBTEP2	Batch restart function executed from data set name (DSN).
ADB2RIP	Revokes impact reporting.
ADB2WCL	Clones a work statement list.
ADB27SPC	Space calculation.
ADBCDCH	Change management deletion enablement.

2.1.9 Product collections

Several collections are also provided with the product and their names and purposes are shown in Table 2-8.

Table 2-8 DB2 Administration Tool collection names

Collection	Purpose
ADBL	Main DB2 Administration Tool collection
ADBCDCH	Change management deletion enablement

2.2 Planning the installation of DB2 Administration Tool

In this section, we describe the elements that need to be considered when installing a new image of DB2 Administration Tool or upgrading an existing installation of DB2 Administration Tool. As a supplement to this section, there is a planning worksheet in “DB2 Administration Tool install and upgrade planning worksheet” on page 500.

When we discuss installing a new image of DB2 Administration Tool, we mean that you are creating a new set of product libraries because you are installing DB2 Administration Tool for the first time or you are installing another iteration of the tool for whatever reason (such as new maintenance iteration you are rolling out gradually).

Although this chapter may seem complex in regard to the steps to install the product, the example was purposefully made to be complex to demonstrate the flexibility of the installation process and cover most of the options. When you actually install the product in your own environments, you discover that many of these steps can be skipped.

When referring to upgrading DB2 Administration Tool, this means applying maintenance to an existing set of tool libraries and then making any necessary product configuration changes.

When referring to upgrading DB2, this means upgrading a release of DB2 with no tool changes. If you are performing both an upgrade of DB2 and installing a new image of the products, the instructions for installing a new image of the product would take precedence. If you are performing both an upgrade of DB2 and an upgrade of the product at the same time, the instructions for upgrading DB2 Administration Tool would also take precedence.

When referring to enabling access to a new DB2 subsystem, this activity refers to enabling an already existing, installed version of DB2 Administration Tool to access a new DB2 subsystem (or data sharing group for data sharing).

2.2.1 Organizational resources

Installing DB2 Administration Tool involves many varied tasks, all of which might not be able to be done by you. It may be necessary to *involve other organizational resources* to complete the installation of the product. Some possible organization resources that might need to be involved are:

- Database administrators

As part of the installation of DB2 Administration Tool, it is necessary to create, alter, and drop objects, bind plans, and packages and grant access to database objects.

- ▶ External security administrators

Installation of the product requires read and write authority to the product libraries, so the appropriate authorities should be given to the SMP/E installer and the tool installer. DB2 Administration Tool authorization switching may require additions to your security environment profiles.

Note: A best practice is to not directly alter the product libraries, but instead create a set of custom libraries to contain any changes made. This way, maintenance does not adversely affect any customization done to the product libraries. If this method is chosen, the installer needs write authority to these custom libraries.

- ▶ z/OS system programmers

Certain items in DB2 Administration Tool require APF authorizations to be added. In many environments, this is the responsibility of the z/OS system programmers, although your organization's environment may vary.

- ▶ SMP/E personnel

Your SMP/E resources manage the initial creation and maintenance applied to the target product libraries and, as such, need to be consulted to manage the naming conventions you use for your target product libraries.

- ▶ TSO administrators

To execute work statement lists in online mode, you might need to have your DB2 libraries as part of your TSO logon procedure if they are not already in the LINKLIST. You might also need to involve these resources to prepare the primary tool CLISTs to be made available.

2.2.2 Product libraries

The first choices to be made when installing or upgrading DB2 Administration Tool pertain to the disposition of the product libraries. It is necessary to create new libraries when installing a new image of the product, but when you are upgrading the products, it is not required, but it might be an attractive option. This section prepares you to help you make an informed choice.

There are questions in “DB2 Administration Tool install and upgrade planning worksheet” on page 500 that pertain to the product libraries.

Installing a new image of DB2 Administration Tool

After the SMP/E target libraries for the product(s) are built, it might be necessary to create copies of these libraries. Some environments allow no access to the original SMP/E libraries. Regardless, a naming convention for both the original and copies of these SMP/E target libraries needs to be determined. A good naming convention should indicate the product, the version, and perhaps another indicator for the maintenance level in the event you want to upgrade using a new set of libraries. An example of a naming convention is shown in 2.3.1, “Product libraries” on page 34.

Important: All libraries used for containing either product modules or custom modules must end with the same data set suffixes provided for by SMP/E. For example, you are provided an EXEC library using a last qualifier of SADBEXEC. Any library used to contain a copy of this information or contain custom information to be used in front of this library *must* end with EXEC. The same holds true for all of the product libraries.

After you determine which libraries should be directly accessed by the product, either the original SMP/E target libraries or their copies, depending on your site standard, these libraries are then referred to as the *operational libraries*.

Upgrading DB2 Administration Tool

If you want to run with multiple versions or even maintenance iterations of the product(s), separate operational libraries need to be maintained, otherwise, you can just apply the maintenance to the already existing SMP/E target libraries and then, if you maintain copies, copy them to your operational libraries.

If you are upgrading to a new version of the product(s), it is assumed that you create a new set of SMP/E target libraries and, if you use them, copies to a new set of operational libraries.

If you are only applying maintenance to the current version of the product(s) and you would still like some systems to be able to use the prior level of maintenance, you also need to create a new set of operational libraries.

Tip: You essentially need a set of libraries for every release and maintenance level of the product that you need to access concurrently.

Customization libraries

Even after the operational libraries are created, the decision should be made as to whether a set of customization libraries should be used to manage changes you might need to make to various product modules. If you make these changes directly against the operational libraries and then apply or copy new maintenance to those libraries, your changes could conceivably be overwritten and lost. The solution to this possible problem is to create a set of *customization libraries*. These libraries can be used to contain changed modules to prevent new maintenance from overwriting them in the operational libraries. There is a mechanism in the product to allocate these customization libraries in front of the product libraries (see 2.3.7, “Preparing the ADBL CLIST” on page 47). The customization libraries need their own naming convention, which is described in 2.3.1, “Product libraries” on page 34.

Note: You only need one set of customization libraries for both DB2 Administration Tool and DB2 Object Comparison Tool.

Upgrading DB2

A new release of the products coincides with every new release of DB2. Every new release of the products should support all currently supported versions of DB2. If you are simply upgrading a version of DB2 that you are accessing (for example, DB2 9 for z/OS to DB2 10 for z/OS) and you already have DB2 Administration Tool V10 installed, there is no need to create separate product libraries and customization libraries for DB2 Administration Tool again.

Enabling access to a new DB2 subsystem

Because the version of DB2 Administration Tool already exists, the product libraries should remain unchanged.

2.2.3 Checkpoint database

The checkpoint database is a number of DB2 objects that are created as part of the installation process and must exist in every subsystem or data sharing group. At least one copy of the checkpoint database must exist per DB2 subsystem or data sharing group that you access with a single version of DB2 Administration Tool.

In addition, if you plan on maintaining multiple versions of DB2 Administration Tool that access the same subsystem, because of changes in the checkpoint database by version, you might need to maintain multiple copies of the checkpoint database in a single subsystem. For example, you want to install DB2 Administration Tool V10, but you want to keep the existing install of DB2 Administration Tool V7.2 operational. If both of these versions of DB2 Administration Tool access the same target subsystem, you need two different checkpoint databases in that target subsystem for the duration of the coexistence of the two versions of the tool.

If you are installing DB2 Administration Tool for the first time, you must create a copy of the checkpoint database in every subsystem you need to access with the tool. If you are installing a new version of the tool, you need to decide whether to upgrade your existing checkpoint database(s) or to create new ones to allow the coexistence of multiple versions of the tool. If you choose to upgrade the checkpoint database, you need to verify that the new structures contained in the latest versions copy of SADBSAMP(ADBCHKPT) match your existing structures. If you do not perform this verification, you might receive bind errors when attempting to bind the plans and packages. Clean binds indicate that the structures are compatible with the new version.

Installing a new image of DB2 Administration Tool

If you are installing a new image of the products, the checkpoint database must be created, either because it does not exist in the subsystem or because the version of the tool has changed so that the checkpoint database has changed. The provided method of creating the objects is to use a DSNTEP2 job containing DDL located in SADBSAMP(ADBCHKPT). You should review this job to make sure the names meet your site naming standards, although there are some suggestions we can make:

- ▶ The checkpoint database default name is ADBDCH and should be named so that the release is indicated in the name (for example, ADBCK10) if you are going to need more than one version or copy of the checkpoint database to coexist in the same subsystem (or data sharing group for data sharing). One image of DB2 Administration Tool interfaces with one copy of the checkpoint database.
- ▶ The table schema should indicate a release-related element (for example, ADBCKT10) if you need to maintain more than one copy of the checkpoint database per subsystem (or data sharing group for data sharing).

Tip: You can use the same database and table schemas for other objects (that is, the change management database and catalog copy version database) but remember that the jobs to edit and submit product object creation have both DROP DATABASE and CREATE DATABASE statements for each of the respective jobs, as the assumption is that each is created in its own database. If this is true, make sure to remove those statements if they are not needed to avoid errors when running the DDL and to prevent data loss when accidentally deleting objects.

- ▶ The DDL to create the checkpoint database contains statements to drop and create a storage group that are used for the database. Make sure to remove these statements if you want to use an already existing storage group.

The reason for these suggestions is primarily to prepare you in the event you need to maintain more than one copy of the checkpoint database.

Upgrading DB2 Administration Tool

If, as part of the upgrade process, you are creating a new set of operational libraries, it is most likely necessary to also have a separate checkpoint database. If you want to keep the current version operational, and the new version contains changes to the checkpoint database structure, you need to create a new copy of the checkpoint database. When upgrading a particular version of DB2 Administration Tool, the SMP/E ++HOLD instructions indicate if the checkpoint database has changed and needs to be updated. ++HOLD is term that is used regarding SMP/E installation and this information can be provided by the person performing the SMP/E installation of the products. The ++HOLD information contains information that might be or is needed after completing a maintenance process (for example, binding of packages, changes to database objects, and so on).

Upgrading DB2

If you are upgrading your level of DB2, you do not need to rebuild or change your already existing checkpoint database, although you may want to rebind the tool plans and packages to receive any benefits by way of the access path changes from the new release of DB2.

Enabling access to a new DB2 subsystem

If the checkpoint database does not exist in the new DB2 subsystem, you must create one as described above.

2.2.4 Segmented temporary table space

If you want to access a DB2 subsystem that is using DB2 8, you must have a segmented table space in a temporary database. If you using a later version of DB2 for z/OS or you know you already have a segmented table space in a temporary database with a buffer pool size of at least 8 KB, you can skip this step.

Installing a new image of DB2 Administration Tool

If you are installing to a DB2 subsystem that is Version 8, the following queries can help you determine if you already have a temporary database and an appropriate segmented table space within a temporary database (see Example 2-1 and Example 2-2).

Example 2-1 Query to find temporary databases

```
SELECT DB.NAME
      FROM SYSIBM.SYSDATABASE DB
      WHERE DB.TYPE = 'T';
```

Example 2-2 Query to find segmented temporary table spaces

```
SELECT TS.DBNAME
      , TS.NAME
      , TS.BPOOL
      FROM SYSIBM.SYSTABLESPACE TS
      , SYSIBM.SYSDATABASE DB
      WHERE DB.NAME = TS.DBNAME
            AND DB.TYPE = 'T'
            AND TS.SEGSIZE > 0;
```

If you already have a temporary database, as shown by the result of the query in Example 2-1 on page 22, you do not need to create another one. If you do not have a temporary database, you need to create one (the provided job has DDL that can be used to create one).

If you already have a segmented table with a buffer pool page size of at least 8 KB within a temporary database, you do not need to create one. Review the query results of the query in Example 2-2 on page 22 to see if there is a table space that uses a buffer pool with a page size of at least 8 KB. If there is one already present, there is no need to create a new one.

Upgrading DB2 Administration Tool

When you are upgrading to a later version of DB2 Administration Tool, you do not need to create a new segmented temporary table space, as you would have created one for the prior release (if you had to).

Upgrading DB2

Upgrading to a later release of DB2 does not require you to create any additional objects.

Enabling access to a new DB2 subsystem

If you are enabling access to a DB2 subsystem that is Version 8, you need to determine if a segmented temporary table space needs to be created in the new subsystem as described above.

2.2.5 Change management database (optional)

The change management database is composed of a number of DB2 objects that are created as part of the installation process and are needed when using change management or storing masks, ignore sets, versions, or version scopes in DB2. In any given DB2 subsystem or data sharing group, you need to create the change management database where you want to use any of these functions. Additionally, if you plan on maintaining multiple versions of DB2 Administration Tool accessing the same subsystem, because of changes in the change management database by version, you might need to maintain multiple copies of the change management database in a single subsystem.

For example, suppose you want to install DB2 Administration Tool V10 but you want to keep the existing installation of DB2 Administration Tool V7.2 operational. If both of these versions of DB2 Administration Tool access the same target subsystem, you need two different change management databases in that target subsystem for the duration of the coexistence of the two versions of the tool.

If you are installing DB2 Administration Tool for the first time, you must create a copy of the change management database in every subsystem you need to access using this tool. If you are installing a new version of the tool, you need to decide whether to upgrade your existing change management database(s) or to create new ones to allow the coexistence of multiple versions of the tool. If you choose to upgrade the change management database, you need to either visually compare the new structures contained in the latest versions copy of SAMP(ADBCHANG) or you can determine if there are differences in the current structures by attempting to bind the new versions product plans and packages and review any errors that result. Clean binds indicate that the structures are compatible with the new version.

Note: Even if you do not plan to use the DB2 Administration Tool change management facility, these objects should still be created. The binding of the plans and packages rely on these database structures being present.

Installing a new image of DB2 Administration Tool

If you are installing a new image of the products, the change management database must be created, either because it does not exist in the subsystem or because the version of the tool has changed so that the change management database has changed. The provided method of creating the objects is using a DSNTEP2 job containing DDL. You should review this job to make sure that the names meet your site naming standards, although there are some suggestions we can make:

- ▶ The change management database's default name is ADBDCHG and should be renamed so that the release is indicated in the name (for example ADBCM10) if you are going to need more than one version or copy of the change management database to coexist in the same subsystem (or data sharing group for data sharing). One image of DB2 Administration Tool interfaces with one copy of the change management database.
- ▶ The table schema used should also indicate a release-related element (for example, ADBCMT10) if you need to maintain more than one copy of the change management database per subsystem (or data sharing group for data sharing).

Tip: You can use the same database and table schemas for other objects (that is, the checkpoint database and catalog copy version database), but remember that the jobs to edit and submit product object creation have both DROP DATABASE and CREATE DATABASE statements for each of the respective jobs, as the assumption is that each is created in its own database, so make sure to remove them if they are not needed to avoid errors when running the DDL or to prevent data loss when accidentally deleting objects.

The reason for these suggestions is primarily to prepare you in the event you need to maintain more than one copy of the change management database.

Upgrading DB2 Administration Tool

If, as part of the upgrade process, you are creating a new set of operational libraries because you want to maintain multiple release of the tools, it is most likely necessary to also have a separate change management database. If you want to keep the current version operational, and the new version contains changes to the change management database structure, you need to create a new copy of the change management database. When upgrading a particular version of DB2 Administration Tool, the provided SMP/E ++HOLD instructions indicate if the change management database has changed and needs to be updated.

Upgrading DB2

If you are upgrading your level of DB2, you do not need to rebuild or change your already existing change management database.

Enabling access to a new DB2 subsystem

If the change management database does not exist in the new DB2 subsystem, you must create one as described above.

2.2.6 Binding the plans and packages

You can maintain multiple sets of plans and collections if you want to be able to run multiple versions of the product in the same subsystem. It is considered a best practice to use meaningful plan and collection names if you want to maintain multiple sets of plans and collections. Detailed examples of these names are shown in 2.3.5, "Binding the plans and packages" on page 44.

Installing a new image of DB2 Administration Tool

When installing DB2 Administration Tool for the first time, you need to bind the plans and packages for the product and, if you plan on keeping multiple sets of plans and collections available, should use meaningful names for the main plan and for each collection.

Upgrading DB2 Administration Tool

If you are simply upgrading to a new release of DB2 Administration Tool, you can simply run the binding of the plans and packages. If you want to maintain a prior release, you need to bind a new set of plans and packages as though this were a new installation.

Upgrading DB2

It is considered a best practice to rebind your plans and packages using the new release of DB2.

Enabling access to a new DB2 subsystem

The same instructions apply here as with a new installation of DB2 Administration Tool. The plans and packages need to be bound to the newly accessed subsystem.

2.2.7 Copying fixed-block to variable-block libraries (optional)

The products are provided in data sets that are fixed-block in nature. Some environments require that their CLIST and EXEC libraries be in the variable-block format. A samp member, SAMP(ADBF2VB), is provided and uses an EXEC to copy the provided fixed-block CLIST and EXEC libraries to variable-block.

Installing a new image of DB2 Administration Tool

If you require variable-block format CLIST and EXEC libraries, you must run the provided EXEC.

Upgrading DB2 Administration Tool

If maintenance affects the CLIST or EXEC libraries, you need to recopy the fixed-block CLIST and EXEC libraries to their variable-block counterparts.

Upgrading DB2

Upgrading to a later release of DB2 does not require you to recreate these libraries.

Enabling access to a new DB2 subsystem

You only need to create variable-block libraries if you create a new set of product libraries.

2.2.8 Preparing the ADBL CLIST

The ADBL CLIST is the sole launch point for DB2 Administration Tool, whether you are using the DB2 Launchpad or going directly into the product. You must determine whether to pass all the necessary information to the product or code it directly into the CLIST. For example, if you want to force the ADBL CLIST to use a particular plan, you can change the line at the beginning of the CLIST from PLAN() to PLAN(<your plan name>). This change forces that iteration of the CLIST to use a plan name matching your provided plan name if a plan name is not passed to it. It is considered a best practice to not modify this module at all and instead pass the necessary parameters to the CLIST, as shown in the example in 2.3.7, “Preparing the ADBL CLIST” on page 47.

Tip: If you decide to provide defaults for other parameters, you might have to maintain multiple copies of the ADBL CLIST even for the same release-level of the DB2 Administration Tool. The best way to avoid having to customize ADBL is to create a CLIST, panel, or EXEC that calls ADBL with the desired parameters.

Installing a new image of DB2 Administration Tool

As described above, you might need to customize a copy of this module.

Upgrading DB2 Administration Tool

If maintenance affects the ADBL CLIST, a re-customization may be necessary.

Upgrading DB2

Because of the DB2 upgrade, you might make direct changes to your copy of ADBL, change the provided default value for DB2LLIBed, or change the specified default value for SECEXIT(). For details about these parameters, refer to the section “Step 8. Customize the ISPF environment” in Chapter 2, “Activating DB2 Admin” in *IBM DB2 Administration Tool for z/OS Version 10 Release 1 User's Guide and Reference*, SC19-3033.

Enabling access to a new DB2 subsystem

The same considerations for upgrading DB2 hold true here. If you are only using the hard coded values relating to DB2 rather than passing the relevant information to the CLIST, you might need another copy of the ADBL CLIST.

2.2.9 Customizing the operating environment on ISPF

The step involves updating the parameters that are used to define the operating environment used by the products. This information is provided by an EXEC that is run and the data entered is stored in a designated Interactive System Productivity Facility (ISPF) table library. This facility also allows you to modify the main DB2 Administration Tool panel to add other menu options. For a detailed example, refer to 2.3.8, “Customizing the operating environment on ISPF” on page 48.

Installing a new image of DB2 Administration Tool

For a first time installation or a maintenance refresh of a new set of libraries, it is necessary to run the EXEC and enter this information. Why do you need to enter this information for a maintenance refresh to a new set of libraries? Keep in mind that created a new product ISPF table library (ISPTLIB or equivalent) and that there is a mechanism provided for copying these stored parameters from one ISPF table library to another, so you do not have to re-enter this information from scratch.

Upgrading DB2 Administration Tool

You do not have to perform any changes to the already stored information

Upgrading DB2

Some of the information that can be entered relates to the DB2 environment and if any of this has changed, you need to update the stored information accordingly.

Enabling access to a new DB2 subsystem

If you have specified DB2 subsystem specific information, you might want to enter the information pertaining to the newly accessed subsystem.

2.2.10 Tailoring the DB2 Selection menu (optional)

You may need to prevent certain DB2 subsystems from being shown in general or by user in an environment. There is a mechanism to control this situation. Keep in mind that DB2 Administration Tool uses your current DB2 security and, if a certain user was not able to perform a particular task before, the product will not allow them to do it now. If you want to tailor the selection of DB2 subsystems, you can either maintain a copy in a separate customization library or you can install a USERMOD to track changes in this module.

Attention: This ability to prevent certain subsystems from being shown is a productivity aid and not a security definition. It will not prevent any user from being able to access any given DB2 subsystem from within the product by using the SYS(XXX) facility, although users are still prevented from performing any task that they cannot already do under existing DB2 security.

Installing a new image of DB2 Administration Tool

You need to modify or create a copy of this module as described above.

Upgrading DB2 Administration Tool

If the SADBPLIB(ADB2DB2X) module has changed because of maintenance, you need to propagate these changes to your modified panel.

Upgrading DB2

Upgrading to a later release of DB2 does not require you to change what has already been done.

Enabling access to a new DB2 subsystem

If this new DB2 subsystem needs its access restricted, you need to modify your panel.

2.2.11 Customizing the ADB2UCUS skeleton

Your environment likely uses a data set naming convention that is different from that used by the provided skeleton SADBSLIB(ADB2UCUS) module. This module needs to be copied and modified in accordance with your environments data set naming conventions.

Installing a new image of DB2 Administration Tool

You need to modify this member (and will need to modify it again as time goes by) for a new image of the DB2 Administration Tool. If you choose to modify members directly inside the libraries to which maintenance is applied, it is a best practice to keep a copy somewhere that will not be impacted by maintenance.

Upgrading DB2 Administration Tool

Unless there are changes to the SADBSLIB(ADB2UCUS) module, there is no need to make any changes. When maintenance is provided, a supplemental member is provided so that this supplemental information can be appended to your existing SADBSLIB(ADB2UCUS) member.

Upgrading DB2

Upgrading to a later release of DB2 does not require you to change what has already been done.

Enabling access to a new DB2 subsystem

There is no requirement to modify this member to enable access to a new DB2 subsystem.

2.2.12 Updating the APF authorization table

Several program and TSO commands must be APF-authorized to use them within the DB2 Administration Tool. All modules provided in SADBLINK must either be copied to an APF-authorized library in the system link list or the operational SADBLINK library must itself be APF-authorized. If you want to use DB2 High Performance Unload (HPU), you must also APF-authorize the main HPU program and either APF-authorize the HPU load library or move the main HPU program load module to another library that is APF-authorized.

Important: If you want to use a set of custom libraries, you need to also APF-authorize the custom copy of SADBLINK if you do not copy the contents of this library to an APF-authorized library.

Installing a new image of DB2 Administration Tool

The tasks described above need to be done for a new installation of the product and also if you are installing another iteration of the product.

Upgrading DB2 Administration Tool

There should be no additional tasks regarding APF-authorization unless a new module is provided in the SADBLINK library.

Upgrading DB2

Upgrading to a later release of DB2 does not require you to change what has already been done.

Enabling access to a new DB2 subsystem

No changes to APF-authorization are needed for the enablement of a new DB2 subsystem.

2.2.13 Preparing to run work statement lists online (optional)

One of the capabilities of the DB2 Administration Tool is that you can run work statement lists online (see Chapter 10, “Work statement lists” on page 259 for an explanation of work statement lists). If you want to invoke DB2 commands, run DB2 utilities or run DB2 High Performance Unload from an online work statement list.

Important: Remember these libraries must be APF-authorized as well.

Installing a new image of DB2 Administration Tool

The tasks described in the introduction must be performed.

Upgrading DB2 Administration Tool

There should be no changes necessary when simply upgrading the products.

Upgrading DB2

You might have to change or add libraries to the system LINKLIST or TSO logon procedure depending on your preferred method of managing the procedure.

Enabling access to a new DB2 subsystem

The DB2 load library for the new DB2 subsystem needs to be added either to the system LINKLIST or the TSO logon procedure as described.

2.2.14 Creating a catalog copy version database (optional)

The catalog copy version database is a number of DB2 objects that are created as part of the installation process and must exist in every subsystem or data sharing group. At least one copy of the catalog copy version database must exist per DB2 subsystem or data sharing group that you access with a single version of DB2 Administration Tool. Additionally, if you plan on maintaining multiple versions of DB2 Administration Tool accessing the same subsystem, because of possible changes in the catalog copy version database by version, you might need to maintain multiple copies of the catalog copy version database in a single subsystem, although there are no structural differences between DB2 Administration V7.2 and DB2 Administration Tool V10. For example, assume you want to install DB2 Administration Tool V10 but want to keep your existing installation of DB2 Administration Tool V7.2 operational. If both of these versions of DB2 Administration Tool access the same target subsystem, you do not require two different catalog copy version databases in that target subsystem for the duration of the coexistence of the two versions of the tool.

If you are installing DB2 Administration Tool for the first time, you must create a copy of the catalog copy version database in every subsystem you need to access with the tool. If you are installing a new version of the tool, you need to decide whether to upgrade your existing catalog copy version database(s) or create new ones to allow the coexistence of multiple versions of the tool. If you choose to upgrade the catalog copy version database, you need to either visually compare the new structures contained in the latest versions copy of SAMP(ADBCCHANG) or you can determine if there are differences in the current structures by attempt to bind the new versions product plans and packages and review any errors that result. Clean binds indicate that the structures are compatible with the new version.

Installing a new image of DB2 Administration Tool

If you are installing a new image of the products, the catalog copy version database must be created, either because it does not exist in the subsystem or because the version of the tool has changed so that the catalog copy version database has changed. The provided method for creating the objects is a DSNTEP2 job containing DDL. You should review this job to make sure the names meet your site naming standards, although there are some suggestions we can make:

- ▶ The catalog copy version database default name is ADBDCC and should be renamed so that the release is indicated in the name (for example, ADBCC10) if you are going to need more than one version or copy of the catalog copy version database to coexist in the same subsystem (or data sharing group for data sharing). One image of DB2 Administration Tool interfaces with one copy of the catalog copy version database.
- ▶ The table schema used should also indicate a release-related element (for example, ADBCCT10) if you need to maintain more than one copy of the catalog copy version database per subsystem (or data sharing group for data sharing).

Tip: You can use the same database and table schemas for other objects (that is, the checkpoint database and change management database), but remember that the jobs to edit and that submit product object creation have both DROP DATABASE and CREATE DATABASE statements for each of the respective jobs, as the assumption is that each is created in its own database, so make sure to remove them if they are not needed to avoid errors when running the DDL or to prevent data loss when accidentally deleting objects.

The reason for these suggestions is primarily to prepare you in the event you need to maintain more than one copy of the catalog copy version database.

Upgrading DB2 Administration Tool

If as part of the upgrade process you create a new set of operational libraries, you most likely need to also have a separate catalog copy version database. If you want to keep the current version operational, and the new version contains changes to the catalog copy version database structure, you need to create a new copy of the catalog copy version database. When upgrading a particular version of DB2 Administration Tool, the provided SMP/E ++HOLD instructions indicate if the catalog copy version database has changed and needs to be updated.

Upgrading DB2

If you are upgrading your level of DB2 to a new version, you need to redefine the catalog copy.

Enabling access to a new DB2 subsystem

If the catalog copy version database does not exist in the new DB2 subsystem, you have to create one as described above.

2.2.15 Creating or updating RUNSTATS views (optional)

DB2 Administration Tool provides the option to allow users to update the RUNSTATS of their own objects in the DB2 catalog. This updating is accomplished by creating views in the DB2 catalog. These views need to be created in every DB2 subsystem or data sharing group in which you want to enable this facility. An example of this updating is shown in 2.3.15, “Creating and updating RUNSTATS views (optional)” on page 59.

Installing a new image of DB2 Administration Tool

A set of these views need to be created in each DB2 subsystem in which they are not already present.

Upgrading DB2 Administration Tool

The views only need to be recreated if maintenance changes the views in the SADBSPAMP(ADBRUNSV) member.

Upgrading DB2

There is no action that needs to be taken if you are upgrading a version of DB2.

Enabling access to a new DB2 subsystem

Again, if these views are not present in the newly enabled DB2 subsystem, they need to be created.

2.2.16 Granting SELECT access to the DB2 catalog

Users of DB2 Administration Tool must have SELECT authority to the DB2 catalog to perform most of the operations in the product. Sample members are provided to grant this authority to PUBLIC. If you do not want to grant this authority to PUBLIC, you need to customize this member to grant the SELECT authority to the appropriate users. If the users of the tool already have SELECT authority, this step can be skipped.

Installing a new image of DB2 Administration Tool

As stated above, SELECT authority to the DB2 catalog, if it is not already there, must be granted to users of the DB2 Administration Tool.

Upgrading DB2 Administration Tool

Applying maintenance to the product does not require the granting of additional authority.

Upgrading DB2

If you upgrade your version of DB2, this new version of DB2 might, and probably will, have additional DB2 catalog tables to which the expected users of the product need SELECT authority.

Enabling access to a new DB2 subsystem

As described in the introduction, make sure all users of DB2 Administration Tool have SELECT access to all the objects in the DB2 catalog.

2.2.17 Additional indexes on DB2 catalog objects (optional)

We suggest that these indexes be created to improve the performance of these queries. Although this is technically an optional task, it is a best practice to create these indexes. The number of indexes created varies with the version of DB2 for the particular subsystem. As of the writing of this book, we recommend 12 additional indexes for DB2 8 and DB2 9 for z/OS and only one additional index for DB2 10 for z/OS. More details about this topic are described in 2.3.17, “Creating additional DB2 catalog indexes (optional)” on page 61.

Installing a new image of DB2 Administration Tool

If you want the performance improvement described above, create the described additional indexes.

Upgrading DB2 Administration Tool

There should be no additional activity required for this step.

Upgrading DB2

The recommended indexes vary by release of DB2, so if you are upgrading to a new release of DB2, consult the current DB2 Administration Tool documentation for the recommended indexes.

Enabling access to a new DB2 subsystem

If you want to get the performance benefit, create the appropriate indexes as described.

2.2.18 Optimizing DSNWZP and DSNZPARM settings

The reverse engineering component of DB2 Administration Tool uses the DSNWZP stored procedure (provided by DB2). You must verify that this stored procedure is operational in any DB2 subsystem on which you want to use the reverse engineering capability of DB2 Administration Tool. If the stored procedure does not complete normally, the reverse engineering component waits for the DSNWZP stored procedure to time out, in which case you might want to verify the DSNZPARM STORTIM(DSN6SYSP) parameter to reduce any excessive wait times due to these time outs.

Installing a new image of DB2 Administration Tool

Verify the status of the DSNWZP stored procedure and the setting of the DSNZPARM STORTIM(DSN6SYSP) parameter as described above.

Upgrading DB2 Administration Tool

There should be no additional activity required for this step.

Upgrading DB2

Verify the status of the DSNWZP stored procedure and the setting of the DSNZPARM STORTIM(DSN6SYSP) parameter as described above.

Enabling access to a new DB2 subsystem

Verify the status of the DSNWZP stored procedure and the setting of the DSNZPARM STORTIM(DSN6SYSP) parameter as described above.

2.2.19 Running the RUNSTATS utility against the DB2 catalog

DB2 Administration Tool accesses the DB2 catalog quite often and if the RUNSTATS utility in the DB2 catalog is not up to date, the performance of the queries the tool uses might not perform optimally. It is a best practice to keep the RUNSTATS on your DB2 catalog relatively current.

Installing a new image of DB2 Administration Tool

Make sure to keep RUNSTATS on your DB2 catalog up to date.

Upgrading DB2 Administration Tool

There should be no additional activity required for this step.

Upgrading DB2

Make sure to keep RUNSTATS on your DB2 catalog up to date.

Enabling access to a new DB2 subsystem

Make sure to keep RUNSTATS on your DB2 catalog up to date.

2.2.20 Tailoring DB2 Administration Tool authorization switching

If enabled, DB2 Administration Tool allows a user to run an alter task even if that user ID does not have the authority to perform changes. As an example, an environment might only allow database administrators to perform changes under one specific global user ID and not under each of their own specific user IDs. There are two ways in the product to accomplish this task:

- ▶ By specifying the global user ID as the value for RUN SQLID
- ▶ By enabling this facility. The change can still be done if the user uses their own ID as well.

Two tool-specific tasks are necessary for the enablement of this facility.

- ▶ The ADBTEPA load module provided in SADBLINK needs to either be copied to an APF-authorized library (which you may have already done in 2.2.12, “Updating the APF authorization table” on page 28) or copied to the LINKLIST.
- ▶ You need to indicate to DB2 Administration Tool where to find this module (which you may have done in 2.2.9, “Customizing the operating environment on ISPF” on page 26). An example of the DB2 Administration Tool steps to enable this function are shown in 2.3.8, “Customizing the operating environment on ISPF” on page 48 and 2.3.12, “Updating the APF authorization table” on page 57.

For specific information about the changes to your security environment, refer to *IBM DB2 Administration Tool for z/OS Version 10 Release 1 User's Guide and Reference*, SC19-3033.

Note: DB2 Administration Tool authorization switching requires that the RRS Attach Facility (RRSAF) of DB2 for OS/390 and z/OS be available.

Installing a new image of DB2 Administration Tool

Copy the ADBTEPX load module and update the operating environment information as described above.

Upgrading DB2 Administration Tool

If load module ADBTEPX has changed as part of the maintenance, it is necessary to recopy this module to the operational libraries.

Upgrading DB2

There should be no additional activity required for this step.

Enabling access to a new DB2 subsystem

There should not be a need to perform any action for this situation, although, depending on how granular your authorization switching security is, you might need to perform some work in your security environment (see the section “Step 22. Tailor DB2 Admin Authorization Switching” in Chapter 2, “Activating DB2 Admin” in *IBM DB2 Administration Tool for z/OS Version 10 Release 1 User's Guide and Reference*, SC19-3033).

2.2.21 Enabling DB2 Administration Tool distributed support

If you want to enable the ability to issue DB2 commands to remote DB2 subsystems, you need to copy the load module to the load library of the default stored procedure address space for the local DB2 subsystem and create a stored procedure in the local DB2 subsystem and bind its related package. An example of this is shown in 2.3.19, “Enabling DB2 Administration Tool distributed support (optional)” on page 62.

Important: Only one load module can exist per DB2 subsystem, but modification to this module usually only happens for new releases of DB2 Admin.

Installing a new image of DB2 Administration Tool

Make sure to copy the load module into the appropriate library and create the stored procedure and bind its associated package.

Upgrading DB2 Administration Tool

If the load module changes, it must be recopied and the stored procedure's package must be rebound.

Upgrading DB2

No action should be necessary.

Enabling access to a new DB2 subsystem

Make sure to copy the load module into the appropriate library and create the stored procedure and bind its associated package.

2.3 Installing a new image of DB2 Administration Tool

In this section, we revisit the majority of the steps referenced in the *IBM DB2 Administration Tool for z/OS Version 10 Release 1 User's Guide and Reference*, SC19-3033 to install the products and put them into the context of an example.

We start with a set of product libraries that contain a single version of DB2 Administration Tool and DB2 Object Comparison Tool. We install a single instance of DB2 Administration Tool and DB2 Object Comparison Tool. The granularity level for an instance of DB2 Administration Tool and DB2 Object Comparison Tool is a single subsystem or data sharing group because they rely on the existence of DB2 catalog objects. The examples include DB2 Object Comparison Tool.

We guide you through the installation process from start to finish with regard to installing a new image of the product. For the example that continues throughout this section, our DB2 environment consists of a single subsystem with DB2 10 for z/OS (Table 2-9). Any differences when installing to a data sharing environment is indicated in the examples and descriptions.

As a supplement to this section, there is a planning worksheet in “DB2 Administration Tool install and upgrade planning worksheet” on page 500. The examples contain portions of this completed worksheet throughout.

Table 2-9 Example environments

Type of environment and identifier	DB2 version
Subsystem DBS1	1015 (V10 NFM)

2.3.1 Product libraries

As mentioned in 2.2, “Planning the installation of DB2 Administration Tool” on page 18, you need to make decisions regarding your product libraries. Before proceeding you should have answered those questions.

Next, we look at the options we chose on the example planning worksheet pertaining to the product libraries (Table 2-10).

Table 2-10 Planning worksheet section pertaining to product libraries

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 1 on page 500.	Do I need to create copies of the original SMP/E target libraries?	No	Yes
2. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries.	None	IBMTOOL.V10ADB00
3. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries.	SADB	SADB
4. See item 4 on page 500.	High-level qualifier for DB2 Object Compare operational libraries.	None	IBMTOOL.V10GOC00
5. See item 5 on page 500.	Prefix of last qualifier for the DB2 Object Compare operational libraries.	SGOC	SGOC
6. See item 6 on page 500.	Am I going to have separate custom libraries?	No	Yes
7. See item 7 on page 500.	High-level qualifier for the customization libraries (if answer for 6 is Yes).	None	IBMTOOL.V10CUS00
8. See item 8 on page 500.	Prefix of last qualifier for the customization libraries (if answer for 4 is Yes).	None	SCUS

Because we want to restrict all access to the original SMP/E target libraries, we copy these SMP/E target libraries to operational libraries using the names we have chosen. We want to use the high-level qualifiers of IBMTOOL.V10ADB00 for the DB2 Administration Tool operational libraries and IBMTOOL.V10GOC00 for the DB2 Object Compare operational libraries. Remember that the *operational libraries* are the original SMP/E target libraries or the copies of the original SMP/E target libraries, as your site standards dictate. These names help us keep these libraries, and their respective maintenance levels, separate from the others.

The high-level example that was chosen indicates both the release level of the product (V10) and the product itself (ADB for Admin, GOC for Object Compare). The two digit number in the second qualifier is available should we want to install a new maintenance level under the same release of the product but be able to access either maintenance level (for example, we roll out the tool maintenance to subsystems across the remainder of our environment). After using SMP/E to create the target libraries, we copy the contents of the target libraries to our operational libraries for DB2 Administration Tool. After this task completes, we have the following libraries:

- ▶ IBMTOOL.V10ADB00.SADBBASE
- ▶ IBMTOOL.V10ADB00.SADBCLST
- ▶ IBMTOOL.V10ADB00.SADBDBRM
- ▶ IBMTOOL.V10ADB00.SADBEXEC
- ▶ IBMTOOL.V10ADB00.SADBLINK
- ▶ IBMTOOL.V10ADB00.SADBLLIB
- ▶ IBMTOOL.V10ADB00.SADBMLIB
- ▶ IBMTOOL.V10ADB00.SADBPLIB
- ▶ IBMTOOL.V10ADB00.SADBSAMP

- ▶ IBMTOOL.V10ADB00.SADBSLIB
- ▶ IBMTOOL.V10ADB00.SADBTLIB

Continuing our example, the following target libraries (or copies) are created for DB2 Object Compare:

- ▶ IBMTOOL.V10GOC00.SGOCBASE
- ▶ IBMTOOL.V10GOC00.SGOCCLST
- ▶ IBMTOOL.V10GOC00.SGOCEXEC
- ▶ IBMTOOL.V10GOC00.SGOCLLIB
- ▶ IBMTOOL.V10GOC00.SGOCMLIB
- ▶ IBMTOOL.V10GOC00.SGOCPLIB
- ▶ IBMTOOL.V10GOC00.SGOCSAMP
- ▶ IBMTOOL.V10GOC00.SGOCSLIB
- ▶ IBMTOOL.V10GOC00.SGOCTLIB

We do not want to modify modules in the operational libraries (to keep our maintenance from overwriting our changes), so we need to determine a high-level data set name for our customization libraries. It is always good to include the general maintenance level in the data set name if you are maintaining multiple library sets.

For this example, we use IBMTOOL.V10CUS00 as the high-level data set name for our customization libraries, but we change the last data set qualifier to be prefaced with SCUS as opposed to SADB to indicate the less tool-specific context of these libraries. We need to create an equivalent custom library for every target library created as part of the SMP/E installation for DB2 Administration Tool (Table 2-2 on page 14). All of these libraries are created empty and can be created with a fraction of their original size, as they never contain all the members (some will always be empty, in fact), but they need to be created because they are allocated as part of the invocation of the DB2 Administration Tool CLIST.

After creating our example custom data sets, we have the following data sets (libraries likely to be empty are indicated):

- ▶ IBMTOOL.V10CUS00.SCUSBASE (empty)
- ▶ IBMTOOL.V10CUS00.SCUSCLST
- ▶ IBMTOOL.V10CUS00.SCUSDBRM (empty)
- ▶ IBMTOOL.V10CUS00.SCUSEXEC
- ▶ IBMTOOL.V10CUS00.SCUSLINK (empty)
- ▶ IBMTOOL.V10CUS00.SCUSLLIB (empty)
- ▶ IBMTOOL.V10CUS00.SCUSMLIB (empty)
- ▶ IBMTOOL.V10CUS00.SCUSPLIB
- ▶ IBMTOOL.V10CUS00.SCUSSAMP
- ▶ IBMTOOL.V10CUS00.SCUSSLIB
- ▶ IBMTOOL.V10CUS00.SCUSTLIB

At this point, all of the target product libraries (or their copies) are populated and the customization libraries are empty. As this example continues, we reference these sample library names.

2.3.2 Creating the checkpoint database

All of the objects that make up the checkpoint database are put into a single DB2 database. The jobs to create the checkpoint database are contained in the SADBSAMP and each has a specific purpose (Table 2-11). You only need to run the one job to create your checkpoint database. It is necessary to create the checkpoint database as many functions, including change management and work statement lists, use it.

Table 2-11 Checkpoint database-related jobs in SADBSAMP for a new installation

Member	Purpose
ADBCHKPT	Creates the checkpoint database objects. Run when you are building the checkpoint database in any given subsystem or data sharing group for the first time or you want to rebuild it from scratch and not retain any data.

The objects built include a database named ADBDCH and a storage group named ADBGCH along with table spaces, table, indexes, foreign keys, and access grants for the various objects (Table 2-12). The user ID under which the objects are created is 'ADB' unless changed. Each table is contained in its own segmented table space.

Table 2-12 Checkpoint database object names

Table space name	Table name	Index name	Foreign key name
ADBSCH	ADBCHKPT	ADBCKPTX	
ADBSPART	ADBPART		ADB1
ADBSCHK	ADBCHK	ADBCHKX1	FKADBCHK
ADBSHOLD	ADBHOLD	ADBHLDX1	FKADBHLD
ADBSBETL	ADBDBETLIST		ADBDBETLIST

The contents of these jobs do not use a proprietary syntax and, technically, everything in the data definition language (DDL) module for creating the checkpoint database can be changed, although we do not recommend the following items be changed:

- ▶ Table space LOCKSIZE: Although it is technically possible to change this item, changing it could cause locking problems if you have many users using the products.
- ▶ Table space SEGSIZE: Again, although it is technically possible to change this item, it is a best practice that the table spaces be built as segmented table spaces, as simple table spaces are deprecated in DB2 10 for z/OS.

Note: If you change the table names or column-related information, you will cause the product to not work correctly.

Note: Remember that the more invasive the modifications you make to the DDL module that makes up the checkpoint database, the higher the cost of your maintenance effort when upgrading DB2 Administration Tool.

Table 2-13 shows the options we chose on the example planning worksheet pertaining to the check point database.

Table 2-13 Planning worksheet section pertaining to the checkpoint database

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 27 on page 501.	Do I want to create a new storage group for my checkpoint database? (Yes/No)	Yes	No
2. See item 28 on page 501.	Checkpoint database storage group name.	ADBGCH	SYSDEFLT
3. See item 10 on page 500.	DB2 subsystem name.	DSN	DBS1
4. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries.	None	IBMTOOL.V10ADB00
5. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries.	SADB	SADB
6. See item 7 on page 500.	High-level qualifier for the customization libraries.	None	IBMTOOL.V10CUS00
7. See item 8 on page 500.	Prefix of last qualifier for the customization libraries.	None	SCUS
8. See item 29 on page 501.	Checkpoint database name.	ADBDCH	TDDCHV10
9. See item 30 on page 501.	Table space name for the ADBCHKPT table.	ADBSCH	ADBSCH
10. See item 31 on page 501.	Table space name for the ADBPART table.	ADBSPART	ADBSPART
11. See item 32 on page 501.	Table space name for the ADBCHK table.	ADBSCHK	ADBSCHK
12. See item 33 on page 501.	Table space name for the ADBHOLD table.	ADBSHOLD	ADBSHOLD
13. See item 34 on page 501.	Table space name for the ADBDBETLIST table.	ADBSBETL	ADBSBETL
14. See item 35 on page 501.	User ID used to create the checkpoint database	ADB	DBAUSER
15. See item 36 on page 501.	Schema name for the checkpoint database tables.	ADBCKTSC	ADBV10
16. See item 37 on page 501.	Schema name for the checkpoint database indexes (if needed to be different from data).	ADBCKXSC	ADBV10

For our example of a clean installation, we make a couple of changes to simulate possible site standards. These changes are:

- Database names begin with 'TD' in the example environment and contain a DB2 Administration release-specific identifier in the event we want to have more than one version of the product active in a single subsystem/data sharing group.

- The schema of all tables has a release-specific identifier (this is a common situation, especially if you run more than one version or maintenance level of the product on a single subsystem/data sharing group).
- We use an already existing STOGROUP for our objects rather than create a new one for the tool installation.

Given these standards, we decide to replace the 'ADB' string that prefaces the database name with 'TD' and add 'V10' to the end so that the database name we use here is 'TDDCHV10', we use a schema name of 'ADBV10' denoting that these objects are used for DB2 Administration Tool V10, and we use a STOGROUP of SYSDEFLT.

We start by installing the product on our DB2 10 for z/OS subsystem, DBS1. Because, as stated earlier, the standard is to not alter any members in the SMPE installation libraries directly, we need to copy the appropriate member from the SMPE installation library (IBMT00L.V10ADB00.SADBSAMP) to our designated custom library (IBMT00L.V10CUS00.SCUSAMP). Because we are creating the checkpoint database for the first time in this subsystem as a clean installation of the products, the job we want to copy is ADBCHKPT, which builds the objects without any unloading or reloading (Table 2-11 on page 37).

To create the checkpoint database objects, we need to modify and run the ADBCHKPT job from the SADBSAMP library. First, we copy the ADBCHKPT member from IBMT00L.V10ADB00.SADBSAMP to the IBMT00L.V10CUS00.SCUSAMP library. We modify the new copy of this member to suit our needs by changing the job card, adding the proper DB2 execution libraries, making sure the DSNTEP2 associated plan name is correct, and then issuing changes to various values:

- Change all occurrences of 'SYSTEM(DSN)' to 'SYSTEM(DBS1)'.
- Change all occurrences of the checkpoint database storage group name from 'ADBGCH' to 'SYSDEFLT'.
- Because we are using an existing storage group rather than creating a new one, we need to comment out the DROP STOGROUP and CREATE STOGROUP statements.
- Change all occurrences of the checkpoint database name from 'ADBDCH' to 'TDDCHV10'.
- Change the SET CURRENT SQLID statement to use 'DBAUSER' instead of 'ADB'.
- Change all occurrences of the checkpoint database table schema from 'ADBCKTSC' to 'ADBV10'.
- Change all occurrences of the checkpoint database index schema from 'ADBCKXSC' to 'ADBV10'.
- Submit the job.

2.3.3 Creating a segmented temporary table space

If you are installing DB2 Administration Tool in a Version 8 subsystem or data sharing group member of DB2 for z/OS, you need to create a segmented table space in a temp database (Table 2-14). DB2 Administration Tool and DB2 Object Compare need this table space to support a declared temporary table that DB2 Administration Tool uses in the course of its operation. Keep in mind that you might already have one created. You can use the queries shown in Example 2-1 on page 22 and Example 2-2 on page 22 to determine if this step needs to be performed.

Table 2-14 Segmented temporary table space-related jobs in SADBSAMP

Member	Purpose
ADBTMPDB	Creates a temporary database and create a segmented table space within it.

Because we are installing DB2 Administration Tool into a DB2 10 for z/OS environment, this step can be skipped.

2.3.4 Change management database

The change management database is made of multiple objects that are used to store various objects used by DB2 Administration Tool change management processes. The schema of these created objects is controlled by setting the SQLID at the beginning of the submitted job. All of the objects that make up the change management database are put into a single DB2 database. The jobs to create the change management database are contained in the SADBSAMP and each has a specific purpose (Table 2-15). You only need to run one of these jobs to create or modify your change management database. It is a best practice to create the change management database, as it is required for the use of the DB2 Administration Tool change management processes.

Table 2-15 Change management database-related jobs in SADBSAMP

Member	Purpose
ADBCCHANG	Creates the change management database objects. Run when you are building the change management database in any given subsystem or data sharing group for the first time or you want to rebuild it from scratch and not retain any data.
ADBCDCH	Creates the view necessary to delete changes from the change management database and bind the package and plan for the delete process. Although this is considered an optional job, it is a best practice to create this view and bind this plan and package, as the continued operation of the change management facility causes these objects to grow.

The objects built include a database named ADBDCHG and a storage group named ADBGCHG, along with table spaces, table, indexes, foreign keys, and access grants for the various objects (Table 2-16). The default schema and owner of the created objects is 'ADB'. Each table is contained in its own segmented table space.

Table 2-16 Change management database object names

Table space name	Table name	Index name	Foreign key name
ADBSCHG	ADBCHG	ADBCHGX1 ADBCHGX2	ADBC_CHG_IGN ADBC_CHG_MSK ADBC_CHG_SVER ADBC_CHG_TVER ADBC_CHG_NVER ADBC_CHG_DVER ADBC_CHG_VSC ADBC_CHG_RCVR
ADBSCHGS	ADBCHGS	ADBCHGSX1 ADBCHGSX2	ADBC_CHGS_CHG
ADBSCHGR	ADBCHGSR	ADBCHGSRX1 ADBCHGSRX2	ADBC_CHGSR_CHG
ADBSCHPRQ	ADBCPREREQ	ADBCPREREQX1	ADBC_PREQ_CHG ADBC_PREQ_CHG2
ADBSCHMSK	ADBCMASK	ADBCMASKX1 ADBCMASKX2	
ADBSCHMSS	ADBCMASKS	ADBCMASKSX1	ADBC_MSKS_MSK
ADBSCHIGN	ADBCIGNORE	ADBCIGNOREX1 ADBCIGNOREX2	
ADBSCHIGS	ADBCIGNORES	ADBCIGNORESX1	ADBC_IGNS_IGN
ADBSCHVER	ADBCVERSION	ADBCVERSIONX1 ADBCVERSIONX2	ADBC_VER_VSC
ADBSCHVLN	ADBCVERLINES	ADBCVERLINESX1 ADBCVERLINESX2	ADBC_VLINS_VER
ADBSCHVSC	ADBCVERSCOPE	ADBCVERSCOPEX1 ADBCVERSCOPEX2	
ADBSCHVSS	ADBCVERSCOPES	ADBCVERSCOPESX1 ADBCVERSCOPESX2	ADBC_VSCS_VSC
ADBSCHID	ADBCID	ADBCIDX1	

Everything in the data definition language (DDL) module for creating the change management database can be changed with the exception of:

- ▶ Table-related items, such as table names and column-related information
- ▶ Table space LOCKSIZE (Although it is technically possible to change this item, it could cause locking problems if you have many users using the products.)
- ▶ Table space SEGSIZE (Again, although it is technically possible to change this item, it is a best practice that the table spaces be built as segmented table spaces, as simple table spaces are deprecated in DB2 10 for z/OS.)

Note: Remember that the more invasive the modifications you make to the DDL module that makes up the checkpoint database, the higher the cost of your maintenance effort when upgrading DB2 Administration Tool.

Table 2-17 lists the options we chose on the example planning worksheet pertaining to the change management database.

Table 2-17 Planning worksheet section pertaining to the change management database

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 44 on page 501.	Do I want to create a new storage group for my change management database? (Yes/No).	Yes	No
2. See item 45 on page 501.	Change management database storage group name.	ADBGCHG	SYSDEFLT
3. See item 10 on page 500.	DB2 subsystem name.	DSN	DBS1
4. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries.	None	IBMTOOL.V10ADB00
5. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries.	SADB	SADB
6. See item 7 on page 500.	High-level qualifier for the customization libraries.	None	IBMTOOL.V10CUS00
7. See item 8 on page 500.	Prefix of last qualifier for the customization libraries.	None	SCUS
8. See item 46 on page 501.	Change management database name.	ADBDCHG	TDDCMV10
9. See item 47 on page 501.	Table space name for the ADBCHG table.	ADBSCHG	ADBSCHG
10. See item 48 on page 501.	Table space name for the ADBCHGS table.	ADBSCHGS	ADBSCHGS
11. See item 49 on page 501.	Table space name for the ADBCHGSR table.	ADBSCHGR	ADBSCHGR
12. See item 50 on page 502.	Table space name for the ADBCPREREQ table.	ADBSCPRQ	ADBSCPRQ
13. See item 51 on page 502.	Table space name for the ADBCMASK table.	ADBSCMSK	ADBSCMSK
14. See item 52 on page 502.	Table space name for the ADBCMASKS table.	ADBSCMSS	ADBSCMSS
15. See item 53 on page 502.	Table space name for the ADBCIGNORE table.	ADBSIGN	ADBSIGN
16. See item 54 on page 502.	Table space name for the ADBCIGNORES table.	ADBSIGS	ADBSIGS
17. See item 55 on page 502.	Table space name for the ADBCVERSION table.	ADBSOVER	ADBSOVER

18. See item 56 on page 502.	Table space name for the ADBCVERLINES table.	ADBSCVLN	ADBSCVLN
19. See item 57 on page 502.	Table space name for the ADBCVERSCOPE table.	ADBSCVSC	ADBSCVSC
20. See item 58 on page 502.	Table space name for the ADBCVERSCOPES table.	ADBSCVSS	ADBSCVSS
21. See item 59 on page 502.	Table space name for the ADBCID table.	ADBSCID	ADBSCID
22. See item 60 on page 502.	User ID used to create the change management database.	ADB	DBAUSER
23. See item 61 on page 502.	Schema name for the change management database tables.	ADBCMTSC	ADBV10
24. See item 62 on page 502.	Schema name for the change management database indexes.	ADBCMISC	ADBV10
25. See item 63 on page 502.	Would you like users to be able to delete cancelled changes from the change management database?	No	Yes

To create the change management database objects, we need to modify and run the ADBCHANG job from the SADBSAMP library. First, we copy the ADBCHANG member from the IBMT00L.V10ADB00.SADBSAMP to the IBMT00L.V10CUS00.SCUSAMP library. We modify the new copy of this member to suit our needs by changing the job card, adding the proper DB2 execution libraries, making sure the DSNTEP2 associated plan name is correct, and then issuing changes to various values as follows:

- ▶ Change all occurrences of 'SYSTEM(DSN)' to 'SYSTEM(DBS1)'.
- ▶ Change all occurrences of the change management database storage group name from 'ADBGCHG' to 'SYSDEFLT'.
- ▶ Because we are using an existing storage group rather than creating a new one, we need to comment out both the DROP STOGROUP and the CREATE STOGROUP statements.
- ▶ Change all occurrences of the change management database name from 'ADBDCHG' to 'TDDCMV10'.
- ▶ Change the SET CURRENT SQLID statement to use 'DBAUSER' instead of 'ADB'.
- ▶ Change all occurrences of the change management database table schema from 'ADBCMTSC' to 'ADBV10'.
- ▶ Change all occurrences of the change management database index schema from 'ADBCMISC' to 'ADBV10'.
- ▶ Submit the job.

Next, we need to enable deletion of cancelled changes from the change management database, so we need to copy and modify the ADBCDC job from SADBSAMP. Copy the ADBCDC member from the IBMT00L.V10ADB00.SADBSAMP library to the IBMT00L.V10CUS00.SCUSAMP library. Change the job card and add the proper DB2 libraries, make sure the DSNTEP2 and DSNTEAD associated plan names are correct, and then make the following changes according to your choices:

- ▶ Change all occurrences of 'SYSTEM(DSN)' to 'SYSTEM(DBS1)'.
- ▶ Change the SET CURRENT SQLID statement to use 'DBAUSER' instead of 'ADB'.

- Change all occurrences of the change management database table schema from 'ADBCMTSC' to 'ADBV10'.
- Submit the job.

2.3.5 Binding the plans and packages

At this stage, we are ready to bind the primary product plans and packages. The members listed in Table 2-18 need to be modified and run based on the installer's choices. This is not the only step where we bind plans and packages for the products. Some plans and packages are bound in 2.3.4, "Change management database" on page 40, 2.3.17, "Creating additional DB2 catalog indexes (optional)" on page 61 and 2.3.19, "Enabling DB2 Administration Tool distributed support (optional)" on page 62.

Table 2-18 Bind-related jobs in SADBSAMP

Member	Purpose
ADBBIND	Binds application packages and plans.

Table 2-19 shows the options we chose on the planning worksheet pertaining to the change management database.

Table 2-19 Planning worksheet section pertaining to the binds of product plans and packages

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries	None	IBMTOOL.V10ADB00
2. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries	SADB	SADB
3. See item 10 on page 500.	DB2 subsystem name	DSN	DBS1
4. See item 7 on page 500.	High-level qualifier for the customization libraries)	None	IBMTOOL.V10CUS00
5. See item 8 on page 500.	Prefix of last qualifier for the customization libraries	None	SCUS
6. See item 36 on page 501.	Schema name for the checkpoint database tables	ADBCKTSC	ADBV10
7. See item 61 on page 502.	Schema name for the change management database tables	ADBCMTSC	ADBV10
8. See item 64 on page 502.	Main Product Plan Name	ADB	PLADBV10
9. See item 65 on page 502.	Main Product Collection Name	ADBL	ADBLV10

We need to bind the rest of our product plans and packages, so now we need to modify and run the ADBBIND job from the SADBSSAMP library. First, we copy the ADBBIND member from IBMT00L.V10ADB00.SADBSSAMP to the IBMT00L.V10CUS00.SCUSAMP library. We modify the new copy of this member to suit our needs by changing the job card, adding the proper DB2 execution libraries, making sure the DSNTIAD associated plan name is correct, and then issuing changes to various values as follows:

- ▶ Change all occurrences of 'ADBA10' to 'IBMT00L.V10ADB00'. This references the correct DBRM library. Make sure to use the high-level product data set and not the high-level customization.
- ▶ Change all occurrences of 'SYSTEM(DSN)' to 'SYSTEM(DBS1)'.
- ▶ Change all occurrences of the package qualifier from 'QUALIFIER(ADBCKTSC)' to 'QUALIFIER(ADBV10)'.
- ▶ Change all occurrences of the package qualifier from 'QUALIFIER(ADBCMTSC)' to 'QUALIFIER(ADBV10)'.
- ▶ In the BIND PLAN statement for the main ADB plan, because we want to enable the deletion of cancelled changes, we need to change the package list for this plan from 'PKLIST (*.ADBL.*)' to 'PKLIST(*.ADBL.* *.ADBCDCH.*)'.

Note: This steps only has to be done for the main plan, which is originally named ADB, and not every plan in the member.

- ▶ Change 'PLAN(ADB)' to 'PLAN(PLADBV10)'. At the time of the writing of this book, there should only be one occurrence of this string.
- ▶ Change all occurrences of 'PACKAGE(ADBL)' to 'PACKAGE(ADBLV10)'.
- ▶ Change all occurrences of '*.ADBL.*' to '*.ADBLV10.*'.
- ▶ Submit the job.

The job may result in a non-zero return code for any of the following reasons:

- ▶ A return code of 4 may occur if you are binding to an earlier release of DB2. This is acceptable.
- ▶ A return code of 8 results if you bind certain packages and you have not created the change management objects (Table 2-16 on page 41). If it is your intention to not use these objects, the return code of 8 on these package binds are acceptable. The packages binds that fail in the change management database are as follows:
 - ADBC DTW
 - ADBC EST
 - ADBC IMU
 - ADBC MRQ
 - ADBC PRE
 - ADBC RCC
 - ADBC RCM
 - ADBC RCP
 - ADBC RCR
 - ADBC SRV
 - ADBC UPC
 - ADBC VCP
 - ADBC VEX
 - ADBC VIC
 - ADBC VSX

2.3.6 Copying fixed-block to variable-block libraries (optional)

If you want to create variable-block versions of the SADBEXEC and SADBCLST libraries, we provide a job to create and populate these data sets.

Table 2-20 Variable block-related jobs in SADBBSAMP

Member	Purpose
ADBFB2VB	Creates variable-block versions of both the fixed-length CLIST and fixed-length EXEC libraries

Table 2-21 shows the options we chose on the example planning worksheet pertaining to the change management database.

Table 2-21 Planning worksheet section pertaining to the creation of the variable-block data sets

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries	None	IBMTOOL.V10ADB00
2. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries	SADB	SADB
3. See item 7 on page 500.	High-level qualifier for the customization libraries)	None	IBMTOOL.V10CUS00
4. See item 8 on page 500.	Prefix of last qualifier for the customization libraries	None	SCUS
5. See item 66 on page 502.	SMP/E Data Set Volume Serial	VLSRNM	TOOL17
6. See item 67 on page 502.	SMP/E Volume Unit Type	SYSALLDA	SYSDA

Additionally, as part of our example, we want to create variable-block CLIST and EXEC libraries, so we need to modify and run the ADBFB2VB job from the SADBBSAMP library. First, we copy the ADBFB2VB member from IBMTOOL.V10ADB00.SADBBSAMP to the IBMTOOL.V10CUS00.SCUSAMP library. We modify the new copy of this member to suit our needs by changing the job card and then issuing changes to various values as follows:

- Change all occurrences of the DB2 Admin Dataset High Level Qual from 'ADBA10' to 'IBMTOOL.V10ADB00'.

Note: Notice that we are copying the contents of the operational libraries to variable-block equivalents. The step to create their empty custom counterparts is shown later in this section.

- Change all occurrences of the default SMP/E Dataset Volume Serial from 'VLSRNM' to 'TOOL17'.
- Change all occurrences of the SMP/E Volume Unit Type from 'SYSALLDA' to 'SYSDA'.

Tip: It is a good idea to use the same volume serial and volume unit type as was used for your original target SMP/E libraries (or their copies).

- Submit the job.

After the job finishes, we have the following additional data sets:

- IBMTOOL.V10ADB00.SADBCLST.VB
- IBMTOOL.V10ADB00.SADBEXEC.VB

Only one more step remains before our product library step is complete. Because we created two new product libraries, we need to create their custom equivalents. Using the method you used earlier to create the earlier set of custom libraries, you should (after creating them) have the following additional data sets (which are empty):

- IBMTOOL.V10CUS00.SCUSCLST.VB
- IBMTOOL.V10CUS00.SCUSEXEC.VB

2.3.7 Preparing the ADBL CLIST

We must prepare the module ADBL in our SADBCLST library for shorthand execution in our environment by moving the module to a library in our CLIST concatenation. A best practice is to not modify the ADBL CLIST. Parameters should instead be passed to the executed CLIST. This CLIST is our primary launch point for both the DB2 Launchpad and the DB2 Administration Tool.

The planning worksheet is listed in Table 2-22.

Table 2-22 Planning worksheet section pertaining to the invocation of the ADBL CLIST

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries	None	IBMTOOL.V10ADB00
2. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries	SADB	SADB
3. See item 4 on page 500.	High-level qualifier for DB2 Object Compare operational libraries	None	IBMTOOL.V10GOC00
4. See item 5 on page 500.	Prefix of last qualifier for the DB2 Object Compare operational libraries	SGOC	SGOC
5. See item 10 on page 500.	DB2 subsystem name	DSN	DBS1
6. See item 7 on page 500.	High-level qualifier for the customization libraries	None	IBMTOOL.V10CUS00
7. See item 8 on page 500.	Prefix of last qualifier for the customization libraries	None	SCUS
8. See item 64 on page 502.	Main Product Plan Name	ADB	PLADBV10
9. See item 61 on page 502.	Schema name for the change management database tables	ADBCMTSC	ADBV10

We want to make the invocation of ADBL as flexible as possible, so we do not make any changes to the product module, but we need to put the IBMTOOL.V10ADB00.SADBCLST.VB library in our CLIST concatenation.

If we wanted to modify this member, we would have copied the ADBL member from the IBMTOOL.V10ADB00.SADBCLST.VB library to the IBMTOOL.V10CUS00.SCUSCLST.VB library, made our modification there, and then added the appropriate library, IBMTOOL.V10CUS00.SCUSLST.VB, to the CLIST concatenation instead.

As mentioned in 2.2.2, “Product libraries” on page 19, the amount of flexibility you need depends on how many disparate subsystems you want to access and how many different versions of the product you plan to run.

If you look at the content of the module, you see parameters that can be set at the beginning of the module. For our example, we are not forcing any of these values by hardcoding them in the CLIST itself. For a detailed discussion of product parameters, see Chapter 3, “Product parameters” on page 65.

Now we can verify whether the initial tailoring effort has been successful by executing the ADBL CLIST using the following TSO command in conjunction with our chosen values:

```
EX 'IBMTOOL.V10ADB00.SADBCLST.VB(ADBL)' 'PLAN(PLADBV10) PROD(IBMTOOL.V10ADB00)
LIBPRE(SADB) PRODADD(IBMTOOL.V10GOC00) LIBAPRE(SGOC) LIST(IBMTOOL.V10CUS00)
LISTPRE(SCUS) CMOWN(ADBV10) SYSTEM(DBS1) VB'
```

The parameters PROD and LIBPRE specify the high level names of the DB2 Administration Tool operation libraries. PRODADD and LIBAPRE specify the high level names of the DB2 Object Compare operational libraries. LIST and LISTPRE specify the high level names of the chosen customization libraries. The 'VB' parameter at the end is necessary if you are using variable-block EXEC and CLIST libraries. If you are not using them, leave this parameter off.

If the command is successful, you should see the DB2 Administration Tool main menu.

Note: If you leave off the SYSTEM(yyyy) parameter, you see a selection list of active DB2 subsystems and members. After a subsystem is selected and Enter is pressed, the DB2 Administration Tool main menu should appear.

2.3.8 Customizing the operating environment on ISPF

For those parameters pertaining to the invocation of the tool that were not set as defaults in the ADBL CLIST, we can change them in this step. Some parameters are set in a general fashion (across all subsystems) and some can be specified at the DB2 subsystem level. This portion of the installation is panel-driven. You invoke the panels to perform this customization through the EXEC(ADB2CUST) module. This module requires no changes prior to execution and should be run directly from the appropriate product library.

Tip: We suggest you run the EXEC(ADB2CUST) module from the library to which you apply maintenance so that you ensure that you are running with the most current version.

Table 2-23 lists the options we chose on the example planning worksheet pertaining to customizing the operating environment on ISPF.

Table 2-23 Planning worksheet section pertaining to customizing the operating environment on ISPF

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries.	None	IBMTOOL.V10ADB00
2. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries.	SADB	SADB
3. See item 95 on page 503.	DB2 subsystem name.	DSN	DBS1
4. See item 96 on page 503.	DB2 remote subsystem location.	None	DBS1
5. See item 12 on page 500.	DB2 runlib library name for the subsystem.	SYS1.DSN.RUNLIB.LOAD	DSN.DB1S.RUNLIB.LOAD
6. See item 13 on page 500.	DB2 exit library name for the subsystem.	SYS1.SDSNEXIT	DSN.DB1S.SDSNEXIT
7. See item 14 on page 500.	DB2 load library name for the subsystem.	SYS1.SDSNLOAD	DSN.DB1S.SDSNLOAD
8. See item 15 on page 500.	DB2 message library name for the subsystem.	None	DSN.DB2VA10.SDSNSPFM
9. See item 16 on page 500.	DB2 panel library name for the subsystem.	None	DSN.DB2VA10.SDSNSPFP
10. See item 17 on page 500.	DB2 CLIST library name for the subsystem.	None	DSN.DB2VA10.SDSNCLST
11. See item 61 on page 502.	Schema name for the change management database tables.	ADBCMTSC	ADBV10
12. See item 78 on page 503.	Schema name for the catalog copy version database tables.	ADBCCTSC	ADBV10
13. See item 83 on page 503.	DB2 Security Exit Type.	STD	STD
14. See item 84 on page 503.	System Identification Method.	None	
15. See item 70 on page 502.	Unit name for batch work data sets.	None	SYSDA
16. See item 2 on page 500.	Unit name for the TSO work data sets.	None	SYSDA
17. See item 85 on page 503.	Installation Name.	None	
18. See item 86 on page 503.	Node Name.	None	

19. See item 87 on page 503.	Will the product be run on a JES3 system?	No	
20. See item 88 on page 503.	UNICODE Translation Technique.	None	
21. See item 23 on page 501.	ISPF Load Library.	None	
22. See item 24 on page 501.	Is the ISPF Load Library in the system LINKLIST?	None	Yes
23. See item 25 on page 501.	ISPF Message Library.	None	SPF.PRODUCT.ISPMLIB
24. See item 26 on page 501.	ISPF Table Library.	None	SPF.PRODUCT.ISPTLIB
25. See item 103 on page 503.	DB2 High Performance Unload enabled?	No	Yes
26. See item 104 on page 504.	DB2 High Performance Unload load library.	None	DSN.HPU410.SINZLINK
27. See item 105 on page 504.	DB2 High Performance Unload panel library.	None	DSN.HPU410.SINZSAMP
28. See item 97 on page 503.	Authorization switching enabled?	No	Yes

For the first step in customizing our operating environment, we invoke the ADB2CUST module from our example operational library IBMT00L.V10ADB00.SADBEXEC by executing the module from a TSO prompt (usually ISPF option 6) using the following command:

EX 'IBMT00L.V10ADB00.SADBEXEC(ADB2CUST)'

The output is the customization main menu (Figure 2-1).

```

DB2 Admin ----- DB2 Administration - Customization ----- 11:27
Option ==>

Options:
  1 - General parameters for DB2 Admin
  2 - DB2 subsystem parameters for DB2 Admin
  3 - Copy general parameters and DB2 subsystem parameters

Library to be used for DB2 Admin customization ISPF tables:
ISPF table library ==> 'IBMT00L.V10CUS00.SCUSTLIB'

```

Figure 2-1 DB2 Administration - Customization main menu

The first step is to designate the ISPF table library where we want our saved changes to go. In this case, we want the changes to be saved to our customization ISPF table library 'IBMTOOL.V10CUS00.SCUSTLIB'. Change the value of the ISPF table library shown to that value. After this task is done, we want to configure our general parameters by entering option 1 and pressing Enter. The customization general parameters panel opens (Figure 2-2). We enter the appropriate values from our worksheet onto this panel.

```
ADB2CUS1 ----- Customization - General Parameters ----- 10:55
Option ==>

General parameters for DB2 Admin:

Press ENTER to save parameters, or END to leave without saving parameters.
More: +

DB2 security exit type ==> STD
(STD,SAMPLE,AUTH,NOCALL,OWN)
Copy version control table qualifier ==> (table owner ID)
System identification method ==> (JESID,SMFID,SYSNAME,NONE)
Unit name for TSO work data sets ==> SYSDA
Unit name for batch data sets ==> SYSDA
Installation name ==>
Node Name ==>
JES3 system ==> (Yes/No)
Unicode translation technique ==> ER ('ER ', etc)
ISPF LLib1(not lnk1st'd) ==>
ISPF LLib2(not lnk1st'd) ==>
ISPF Message Library ==> SPF.PRODUCT.ISPMLIB
ISPF Table Library ==> SPF.PRODUCT.ISPTLIB
DB2 Admin APF Library ==>

Current SYSAFF information for this MVS system:
JES ID: NONE (found via JESID method)
SMF ID: SY4A (found via SMFID method)
MVS system name: STPL (found via SYSNAME method)
```

Figure 2-2 Example Customization - General Parameters panel

Note that a couple of fields were left blank. The copy version control table qualifier is left blank because we will manage this field at the subsystem level in a later step. The DB2 admin APF library was left blank because the modules are being moved to another library in the system link list (see 2.2.12, "Updating the APF authorization table" on page 28 and 2.3.12, "Updating the APF authorization table" on page 57).

After the information is entered, return to the DB2 Administration Tool customization main menu, press 2, and then press Enter. The DB2 subsystem customization main menu opens (Figure 2-3). If this is the first time you have opened this panel using the chosen ISPF table library, you get a message stating that ADB2DB2D was not found in the library. This is normal.

```
ADB2CUS2 ----- Customization - DB2 Subsystem Parameters ----- 12:04
Option ==>

Options:
  1 - Merge list of active DB2 subsystem(s) into ISPF table
  2 - Edit ISPF table

Currently active DB2 subsystem(s) on this MVS system:
DB2 subsystem(s):   DBS1
Data sets to be used for DB2 Admin customization:
ISPF table library: 'IBMT00L.V10CUS00.SCUSTLIB'
```

Figure 2-3 DB2 subsystem customization main menu

Because this is the first time you are entering these subsystem parameters, choose option 1 and press Enter. You get a notification regarding how many subsystems have been added to the ISPF table. This message is informational. Next, we need to enter the more specific DB2 subsystem information. Now, from the DB2 subsystem customization main menu, choose option 2 and press Enter. An edit session opens that allows you to enter specific information into the table (Example 2-3).

Example 2-3 Initial example of the DB2 subsystem customization edit session

```
*
* Tags recognized by ADB2CUST EXEC:
* -----
* :nick.      Name of DB2 subsystem or DB2 group being described
* :desc.      Text to be displayed for this DB2 subsystem or DB2 group
* :secexit.   DB2 security exit type (STD,SAMPLE,AUTH,OWN,NOCALL)
* :cvtown.    Copy version table owner
* :stcnam.    Started task name of this DB2 subsystem
* :grpnam.    DB2 group name of this DB2 subsystem or DB2 group
* :jclass.    Job class to be used for batch DB2 utility jobs
* :jsysaff.   SYSAFF to be used for batch DB2 utility jobs
* :instparm.  Installation name
* :utilpre.   Utility data set prefix (USERID,OWNER,CREATEDBY,name)
* :ssid.      Subsystem name of the remote subsystem (blank if local)
* :nodenam.   Node name of the remote subsystem (blank if local)
* :locnam.    Location name of the remote subsystem (blank if local)
* :authsw.    Authorization Switching enabled (YES,NO)
* :newappl.   ISPF application ID
* :prompt.    Default value for Prompt Options (YES,NO)
* :resetopt.  Default value for Reset to Default at Startup (YES,NO)
* :nstuproc.  Number of job steps in the DSNUPROC procedure
* :cmown.     Owner (Qualifier) of Change Management database objects
* :swssid.    Allow switch of SSID
* :authswid.  DB2 Security ID to use for auth-switching
* :concent.   DB2 CONCENTRATE STATEMENTS WITH LITERALS (YES,NO)
```

```

* :currcom. DB2 Use CONCURRENT clause on SQL (YES,NO)
* :hpu.      High Performance Unload (HPU) enabled (YES,NO)
* :hpulib.   High Performance Unload (HPU) load library
* :hpuplib.  High Performance Unload (HPU) parmlib library
* :uexelib.  REXX user exit library
*
* :bl2llib.  DB2 load library
* :bl2rlib.  DB2 run library for sample program DSNTIAUL
* :bl2mlib.  DB2 message library
* :bl2plib.  DB2 panel library
* :bl2slib.  DB2 skeleton library
* :bl2tlib.  DB2 table library
* :bl2elib.  DB2 REXX EXEC library
* :bl2clib.  DB2 CLIST library
*
* Main menu option tags (prefixed by letter a-j)
*
* :aopt.     Option
* :adescr.   Option description
* :aispf.    ISPF statement for option
* :apan.     ISPF panel for option
* :asql.     SQL statement for option
* :acmd.     DB2 Admin command for option
* :anewat.   New DB2 attachment (YES,NO)
*

```

```
:nick.DBS1 :desc.(No description for this DB2 subsystem)
```

```
* File created on 23 Jan 2011 12:16:38 by DBAUSER
```

You see that a line was added when you merged the list of active DB2 subsystems with the ISPF table. The `.nick` keyword defines which DB2 subsystem to which that information pertains. You can use `.nick.*` to designate generic or common elements. You see comments at the beginning of the member showing the available keywords. We now enter information for both the common and subsystem-specific parameters. After using the information from our worksheet, we end up with two sections (Example 2-4).

Example 2-4 Sections for the common and subsystem-specific parameters

```

*
* Tags recognized by ADB2CUST EXEC:
* -----
* :nick.     Name of DB2 subsystem or DB2 group being described
* :desc.     Text to be displayed for this DB2 subsystem or DB2 group
* :secexit.  DB2 security exit type (STD,SAMPLE,AUTH,OWN,NOCALL)
* :cvtown.   Copy version table owner
* :stcnam.   Started task name of this DB2 subsystem
* :grpnam.   DB2 group name of this DB2 subsystem or DB2 group
* :jclass.   Job class to be used for batch DB2 utility jobs
* :jsysaff.  SYSAFF to be used for batch DB2 utility jobs
* :instparm. Installation name
* :utilpre.  Utility data set prefix (USERID,OWNER,CREATEDBY,name)
* :ssid.     Subsystem name of the remote subsystem (blank if local)
* :nodenam.  Node name of the remote subsystem (blank if local)
* :locnam.   Location name of the remote subsystem (blank if local)
* :authsw.   Authorization Switching enabled (YES,NO)

```

```

* :newappl. ISPF application ID
* :prompt. Default value for Prompt Options (YES,NO)
* :resetopt. Default value for Reset to Default at Startup (YES,NO)
* :nstuproc. Number of job steps in the DSNUPROC procedure
* :cmown. Owner (Qualifier) of Change Management database objects
* :swssid. Allow switch of SSID
* :authswid. DB2 Security ID to use for auth-switching
* :concent. DB2 CONCENTRATE STATEMENTS WITH LITERALS (YES,NO)
* :currcom. DB2 Use CONCURRENT clause on SQL (YES,NO)
* :hpu. High Performance Unload (HPU) enabled (YES,NO)
* :hpulib. High Performance Unload (HPU) load library
* :hpuplib. High Performance Unload (HPU) parmlib library
* :uexelib. REXX user exit library
*
* :bl2llib. DB2 load library
* :bl2rlib. DB2 run library for sample program DSNTIAUL
* :bl2mlib. DB2 message library
* :bl2plib. DB2 panel library
* :bl2slib. DB2 skeleton library
* :bl2tlib. DB2 table library
* :bl2elib. DB2 REXX EXEC library
* :bl2clib. DB2 CLIST library
*
* Main menu option tags (prefixed by letter a-j)
*
* :aopt. Option
* :adescr. Option description
* :aispf. ISPF statement for option
* :apan. ISPF panel for option
* :asql. SQL statement for option
* :acmd. DB2 Admin command for option
* :anewat. New DB2 attachment (YES,NO)
*

:nick.* :desc.General Parameters
       :newappl.ADB
       :hpu.NO
       :hpulib.'DSN.HPU410.SINZLINK'
       :hpuplib.DSN.HPU410.SINZSAMP
       :aopt.I
       :adescr.DB2I
       :aispf.SELECT CMD(%DSNECPRI SSID(&DB2SYS)) NEWAPPL(DSNE) PAS
       :copt.C
       :cdescr.DB2 Object Comparison Tool
       :cpan.GOCMENU .*

:nick.DBS1 :desc.DB2 Version 10 NFM
          :cvtown.ADB
          :jclass.A
          :locnam.DBS1
          :authsw.YES
          :newappl.ADB1
          :hpu.YES
          :bl2llib.'DSN.DSNA.SDSNEXIT'
                  'DSN.DSNA.SDSNLOAD'

```

```
:b12r1ib.'DSN.DSNA.RUNLIB.LOAD'  
:b12m1ib.'DSN.DB2VA10.SDSNSPFM'  
:b12p1ib.'DSN.DB2VA10.SDSNSPFP'  
          'DSN.DB2VA10.SDSNPFPE'  
:b12c1ib.'DSN.DB2VA10.SDSNCLST'  
:copt.C  
:cdescr.DB2 Object Comparison Tool  
:cpan.GOCMENU
```

* File created on 9 Feb 2011 10:35:02 by DBAUSER

Note that we have specified several parameters as generic. These options will be enacted for every subsystem accessed. The parameters we specified as generic are:

► desc

This parameter is for documentation purposes. It is shown on the subsystem selection panel.

► newappl

This parameter has been specified as the default in the event that no newappl is specified for each DB2 subsystem. It is a good idea to use a different newappl value for each DB2 subsystem to prevent inadvertent reuse of ISPF variables due to a ISPF variable pool overlap.

► hpu

This is also specified as a default.

► hpullib and hpuplib

These libraries are specified as a default. If you plan on using different HPU libraries for each DB2 subsystem, you can specify the libraries that are different in each subsystem.

► copt, cdescr, and cpan

We have added the option to invoke DB2 Object Compare. copt is the selection letter that will appear on the main DB2 Administration Tool main menu (we are using 'C' in this example). cdescr is the descriptive text displayed next to the selection letter. Lastly, cpan is the ISPF panel used to invoke the product specified.

All options that are specified in the generic options will be overridden if they are also specified in any given section for a specific DB2 subsystem.

Regarding the options we specified for the specific DB2 subsystem (DBS1), the following options are as follows:

► desc

As above, the desc field is purely for documentation purposes.

► cvtown

This is the catalog copy version table schema. If you do not want to use these objects, this parameter does not need to be provided.

► jclass

This is the batch job class for generated batch jobs when the user does not specify one.

► locnam

This is the location name of the DB2 subsystem.

- ▶ authsw
A parameter indicating whether auth-switching is enabled.
- ▶ newappl
This is the newappl ID we use for this DB2 subsystem.
- ▶ hpu
This is an indicator as to whether to enable use of DB2 High Performance Unload (HPU). If you do not have this product or do not want to enable it for use with DB2 Administration Tool, you can either leave this parameter off or specify NO.
- ▶ bl2llib, bl2rllib, bl2mlib, bl2plib, and bl2clib
These are the DB2-specific libraries for this DB2 subsystem.
- ▶ aopt, adescr, and apan
For this example, we enable the ability to launch SPUFI from the main DB2 Administration Menu. aopt, adescr, and apan are the selection letter, the descriptive text displayed next to the selection letter, and the ISPF command used to invoke the program, respectively.
- ▶ copt, cdescr, and cpan
This is the same information we specified in the generic section and is redundant. If you choose, you could remove this information from the DB2-subsystem specific area or from the generic area.

You might have noticed that we used aopt, adescr, and apan and then copt, cdescr, and cpan. We used this setup in the example to show how to put a blank line on the DB2 Administration Tool main menu because we did not use bopt, bdescr, and bpan.

You can now exit the edit session panel.

2.3.9 Tailoring the DB2 Administration Tool main menu

This step allows us to add options to the main DB2 Administration Tool main menu. The primary way to invoke the DB2 Object Comparison Tool is by using the DB2 Administration Tool main menu. We do not directly tailor any of the panels for this example because we added SPUFI and DB2 Object Compare as part of the example in 2.3.8, “Customizing the operating environment on ISPF” on page 48.

2.3.10 Tailoring the DB2 Selection menu

You can control the default DB2 subsystem or hide DB2 subsystems from the user by modifying the ADB2DB2X panel. The example logic in this panel shows how we control which DB2 subsystems are hidden or used as defaults by the user and user prefix. We do not tailor this panel as part of this example. For more information about this topic, refer to section “Step 9(b). Specify DB2 subsystems on the DB2 Selection Menu” in Chapter 2 “Activating DB2 Admin” in *IBM DB2 Administration Tool for z/OS Version 10 Release 1 User's Guide and Reference*, SC19-3033.

2.3.11 Customizing the ADB2UCUS skeleton

In this step, you set up how data set names are managed (if you are not using templates). It is usually a necessary to modify this member in some way as the default naming conventions we provide seldom match the more specific naming conventions that would be used in a corporate environment. We provide some basic information regarding how to customize this module, but no specific examples. For specifics regarding what options you can use to customize this module, refer to section “Step 10. Customize the ADB2UCUS” in Chapter 2, “Activating DB2 Admin” in *IBM DB2 Administration Tool for z/OS Version 10 Release 1 User's Guide and Reference*, SC19-3033.

In anticipation of the changes that will need to be made to this module, copy the ADB2UCUS member from IBMT00L.V10ADB00.SADBSLIB to the IBMT00L.V10CUS00.SCUSLIB library. We modify the new copy of this member to suit our needs by making changes to various values.

2.3.12 Updating the APF authorization table

Certain facets of the DB2 Administration Tool must be APF-authorized for them to work properly. This work is most likely carried out by your z/OS system programmers (your site may vary). All of the modules provided in the SADBLINK library need to either be copied to a library that is APF-authorized or the SADBLINK library itself can be APF-authorized¹. In addition, two programs and TSO commands must be invoked as authorized: ADB2ATH and ADB2UTIL. This task is accomplished by modifying SYS1.PARMLIB(IKJTSoxx) according to the instructions contained in section “Step 12. Update the APF Authorization table” of Chapter 2, “Activating DB2 Admin” in the *IBM DB2 Administration Tool for z/OS Version 10 Release 1 User's Guide and Reference*, SC19-3033.

Additionally, if you intend to use DB2 High Performance Unload, you also need to either copy the contents of your SINZLINK library to another APF-authorized library or APF-authorize the SINZLINK library.

2.3.13 Preparing to run work statement lists online

In order for users to be able to run work statement lists online, the DB2 load library must be made accessible by either adding it to the system LINKLIST or it must be added to the TSO logon procedure used prior to invoking the product. For more information about this task, refer to section “Step 13. Prepare to run a work statement online” of Chapter 2, “Activating DB2 Admin” of *IBM DB2 Administration Tool for z/OS Version 10 Release 1 User's Guide and Reference*, SC19-3033.

Additionally, if you intend to use DB2 High Performance Unload (HPU), depending on whether you copied the SINZLINK modules to another library or you APF-authorized the SINZLINK library itself, you need to add the HPU SINZLINK library or the library to which you copied the SINZLINK modules to either the system LINKLIST or to the TSO logon procedure used prior to invoking the product. This task is only necessary if you intend to use HPU in a work statement list that is run in online mode.

¹ This APF-authorized library must either be in the system link list, or must be registered as the “DB2 Admin APF Library” in the customization panel ADB2CUS1 (see Figure 2-2 on page 51).

2.3.14 Creating a catalog copy version file (optional)

In some environments, it is useful to restrict access to the actual DB2 catalog. DB2 Administration Tool provides a way to switch between operating against a copy of the DB2 catalog and operating against an actual DB2 catalog. To enable the use of DB2 catalog copies, you need to create the repository used to keep track of these copies with the job shown in Table 2-24.

Table 2-24 Catalog copy-related jobs in SADBSAMP

Member	Purpose
ADBCATVT	Creates a temporary database and create on segmented table space within it.

Table 2-25 shows the options we chose on the example planning worksheet pertaining to the catalog copy version database.

Table 2-25 Planning worksheet section pertaining to the catalog copy version database

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries.	None	IBMTOOL.V10ADB00
2. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries.	SADB	SADB
3. See item 10 on page 500.	DB2 subsystem name.	DSN	DBS1
4. See item 7 on page 500.	High-level qualifier for the customization libraries.	None	IBMTOOL.V10CUS00
5. See item 8 on page 500.	Prefix of last qualifier for the customization libraries.	None	SCUS
6. See item 73 on page 502.	Do I want to create a new storage group for my catalog copy version database? (Yes/No)	Yes	No
7. See item 74 on page 502.	Catalog copy version database storage group name.	ADBGCC	SYSDEFLT
8. See item 75 on page 502.	Catalog copy version database name.	ADBDC	TDDCCV10
9. See item 76 on page 503.	Table space name for the ADBCATVT table.	ADBSCC	ADBSCC
10. See item 77 on page 503.	User ID used to create the catalog copy version database.	ADB	DBAUSER
11. See item 78 on page 503.	Schema name for the catalog copy version database tables.	ADBCCTSC	ADBV10
12. See item 79 on page 503.	Schema name for the catalog copy version database indexes.	ADBCCXSC	ADBV10

Next, we need to bind the rest of our product plans and packages, so now we need to modify and run the ADBCATVT job from the SADBSAMP library. We copy the ADBCATVT member from IBMTOOL.V10ADB00.SADBSAMP to the IBMTOOL.V10CUS00.SCUSAMP library. We modify the new copy of this member to suit our needs by changing the job card, adding the proper DB2 execution libraries, making sure the DSNTEP2 associated plan name is correct, and then issuing changes to various values as follows:

- ▶ Change all occurrences of 'SYSTEM(DSN)' to 'SYSTEM(DBS1)'.
- ▶ Change the SET CURRENT SQLID statement to use 'DBAUSER' instead of 'ADB'.
- ▶ Change all occurrences of 'ADBGCC' to 'SYSDEFLT'.
- ▶ Change all occurrences of 'ADBDC' to 'TDDCCV10'.
- ▶ Change all occurrences of 'ADBCCTSC' to 'ADBV10'.
- ▶ Change all occurrences of 'ADBCCXSC' to 'ADBV10'.
- ▶ You will need to comment out the DROP STOGROUP and CREATE STOGROUP statements because you are using a storage group that already exists.
- ▶ Submit the job.

2.3.15 Creating and updating RUNSTATS views (optional)

If you would like users to be able to update RUNSTATS information for their own objects, it is necessary to create views to enable this option. The facility to update RUNSTATS information in DB2 Administration Tool is the 'UR' line command (Table 2-26).

Table 2-26 RUNSTATS views-related jobs in SADBSAMP

Member	Purpose
ADBRUNSV	Creates views to allow object creators to update the RUNSTATS information of their own objects in the DB2 catalog.

Table 2-27 shows the options we chose on the example planning worksheet pertaining to the creation of the RUNSTATS views.

Table 2-27 Planning worksheet section pertaining to the creating RUNSTATS update views

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries	None	IBMTOOL.V10ADB00
2. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries	SADB	SADB
3. See item 10 on page 500.	DB2 subsystem name	DSN	DBS1
4. See item 7 on page 500.	High-level qualifier for the customization libraries)	None	IBMTOOL.V10CUS00
5. See item 8 on page 500.	Prefix of last qualifier for the customization libraries	None	SCUS
6. See item 82 on page 503.	RUNSTATS view schema name	RUNSTATS	ADBV10

We want to create views so that users can update RUNSTATS for their own objects, so we need to create views by modifying and running the ADBRUNSV job from the SADBSAMP library. First, we copy the ADBRUNSV member from IBMTOOL.V10ADB00.SADBSAMP to the IBMTOOL.V10CUS00.SCUSAMP library. We modify the new copy of this member to suit our needs by changing the job card, adding the proper DB2 execution libraries, making sure the DSNTEP2 associated plan name is correct, and then issuing changes to various values as follows:

- ▶ Change all occurrences of 'S(DSN)' to 'S(DBS1)'.
- ▶ Change all occurrences of the view schema name from 'RUNSTATS.' (period is included to avoid updating the leading comments) to 'ADBV10.' (period needs to be included here as well).
- ▶ Submit the job.

2.3.16 Granting SELECT access to the DB2 catalog

To be able to work with the content of the DB2 catalog within the product, users must have SELECT authority. You can tailor the provided jobs to make sure you only give SELECT authority to those users who you truly want to have it. If it is not important to control this access at the individual user level, you can give SELECT access to the PUBLIC user. Table 2-28 shows the jobs that can be run to grant this SELECT authority.

Table 2-28 Grant-related jobs in SADBSAMP

Member	Purpose
ADBGC	Grants SELECT authority to the DB2 catalog for DB2 versions 8 and 9.
ADBGC10	Grants SELECT authority to the DB2 catalog for DB2 version 10.

Table 2-29 shows the options we chose on the example planning worksheet pertaining to granting SELECT authority.

Table 2-29 Planning worksheet section pertaining to the granting of SELECT authority on the DB2 catalog

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries	None	IBMTOOL.V10ADB00
2. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries	SADB	SADB
3. See item 10 on page 500.	DB2 subsystem name	DSN	DBS1
4. See item 7 on page 500.	High-level qualifier for the customization libraries)	None	IBMTOOL.V10CUS00
5. See item 8 on page 500.	Prefix of last qualifier for the customization libraries	None	SCUS

We need to grant SELECT authority to the DB2 catalog to users of DB2 Administration Tool, so we need to modify and run either the ADBGC member or the ADBGC10 member, depending on the version of the DB2 subsystem, from the SADBSAMP library. Because our example environment is a DB2 subsystem that is V10 NFM, we copy the ADBGC10 member from IBMT00L.V10ADB00.SADBSAMP to the IBMT00L.V10CUS00.SCUSAMP library. We modify the new copy of this member to suit our needs by changing the job card, adding the proper DB2 execution libraries, verifying the DSNTIAD plan name, and then issuing changes to various values as follows:

- ▶ Change all occurrences of 'S(DSN)' to 'S(DBS1)'.
- ▶ Submit the job.

2.3.17 Creating additional DB2 catalog indexes (optional)

Some DB2 catalog tables are accessed in ways where the DB2-provided indexes are not used. Although this step is not technically required, performance can be enhanced considerably by modifying and running the provided job (Table 2-30) to create these indexes.

Table 2-30 SDB2 catalog index-related jobs in SADBSAMP

Member	Purpose
ADBCX	Creates additional indexes on the DB2 catalog for DB2 8 and 9.
ADBCX10	Creates additional indexes on the DB2 catalog for DB2 10.

Table 2-31 shows the options we chose on the example planning worksheet pertaining to the creation of additional DB2 catalog indexes.

Table 2-31 Planning worksheet section pertaining to the creation of additional indexes on the DB2 catalog

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries	None	IBMT00L.V10ADB00
2. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries	SADB	SADB
3. See item 10 on page 500.	DB2 subsystem name	DSN	DBS1
4. See item 7 on page 500.	High-level qualifier for the customization libraries)	None	IBMT00L.V10CUS00
5. See item 8 on page 500.	Prefix of last qualifier for the customization libraries	None	SCUS

We need to create the additional indexes on the DB2 catalog, so we need to modify and run the ADBCX10 member from the SADBSAMP library. We copy the ADBCX10 member from IBMT00L.V10ADB00.SADBSAMP to the IBMT00L.V10CUS00.SCUSAMP library. We modify the new copy of this member to suit our needs by changing the job card, adding the proper DB2 execution libraries, verifying the DSNTIAD plan name, and then issuing changes to various values as follows:

- ▶ Change all occurrences of 'SYSTEM(DSN)' to 'SYSTEM(DBS1)'.
- ▶ Submit the job.

2.3.18 Running the RUNSTATS utility against the DB2 catalog (optional)

It is best practice to make sure that the RUNSTATS utility in your DB2 catalog remains relatively current. As stated in 2.2.19, “Running the RUNSTATS utility against the DB2 catalog” on page 32, the product makes extensive use of the DB2 catalog and inaccurate statistics can result in degraded performance. The statistics should be updated in the DB2 catalog. Packages would need to be rebound to benefit from any changes to the access path.

2.3.19 Enabling DB2 Administration Tool distributed support (optional)

DB2 Administration Tool uses a stored procedure to run DB2 commands that is run when connected to a remote site. This step creates and binds this stored procedure. The job that needs to be copied and modified is shown in Table 2-32.

Table 2-32 Distributed support-related jobs in SADBSAMP

Member	Purpose
ADBRCPC	Binds the ADB2RCM member in the ADBL package and create the stored procedure.

Table 2-33 shows the options we chose on the example planning worksheet pertaining to the enablement of remote DB2 command execution.

Table 2-33 Planning worksheet section pertaining to enabling DB2 Administration Tool distributed support

Item No.	Item	Starting or default value (if applicable)	Chosen value
1. See item 2 on page 500.	High-level qualifier for DB2 Administration Tool operational libraries	None	IBMTOOL.V10ADB00
2. See item 3 on page 500.	Prefix of last qualifier for the DB2 Administration Tool operational libraries	SADB	SADB
3. See item 10 on page 500.	DB2 subsystem name	DSN	DBS1
4. See item 7 on page 500.	High-level qualifier for the customization libraries)	None	IBMTOOL.V10CUS00
5. See item 8 on page 500.	Prefix of last qualifier for the customization libraries	None	SCUS
6. See item 65 on page 502.	Main Product Collection Name	ADBL	ADBLV10
7. See item 81 on page 503.	Schema name for the stored procedure to execute DB2 commands when connected to a remote site	ADB	ADBV10
8. See item 80 on page 503.	Workload Manager (WLM) environment name	DSNWLM1	WLMA

We want to enable the ability to run DB2 commands when connected to a remote site by creating and binding the stored procedure, so we need to modify and run the ADB2RCPC job from the SADBSSAMP library. We copy the ADB2RCPC member from IBMT00L.V10ADB00.SADBSSAMP to the IBMT00L.V10CUS00.SCUSSSAMP library. We modify the new copy of this member to suit our needs by changing the job card, adding the proper DB2 execution libraries, verifying the DSNTIAD plan name, and then issuing changes to various values as follows:

- ▶ Change all occurrences of 'SYSTEM(DSN)' to 'SYSTEM(DBS1)'.
- ▶ Change all occurrences of 'PACKAGE(ADBL)' to 'PACKAGE(ADBLV10)'.
- ▶ Change all occurrences of 'COLLID ADBL' to 'COLLID ADBLV10'.
- ▶ Change all occurrences of 'ADBL.' (a period is included to avoid updating other items) to 'ADBLV10.' (a period also needs to be included here).
- ▶ Change all occurrences of the stored procedure schema name from 'ADB.' (a period is included to avoid updating the leading comments) to 'ADBV10.' (a period needs to be included here as well).
- ▶ Change all occurrences of 'DSNWLM1' to 'WLMA'.
- ▶ Submit the job.



Product parameters

This chapter discusses the product parameters used to control DB2 Administration Tool behavior at a general, subsystem, and user level.

This chapter contains the following topics:

- ▶ Introduction to product parameters
- ▶ Installation and customization parameters
- ▶ General operational parameters
- ▶ Change management ID parameters

3.1 Introduction to product parameters

DB2 Administration Tool and DB2 Object Comparison Tool both use various parameters to manage how the product operates at an installation level and a user level. Some of these parameters are necessary for the basic operation of the product and some of them manage how the product behaves according to user preferences.

3.2 Installation and customization parameters

In this section, we discuss the parameters that are specified that define the general function of the tools. These parameters are needed only to make the product operational, are not used by the individual users, and there is no method within the product to change these parameters.

The parameters can be specified at several levels and there is a hierarchy to how a certain value is derived. For each installation and customization parameter, the tool searches the following sources in sequence:

1. The parameter is passed to the ADBL CLIST at execution time.
2. The parameter value is specified at the DB2 subsystem-specific level (Example 3-1 on page 68).
3. The parameter value is specified as a default DB2-subsystem customization option (Example 3-1 on page 68).
4. The parameter value is specified on the general customization panel (Figure 3-2 on page 67).

Every parameter is not necessarily able to be designated in every one of these sources. The first parameter from the first source in which it was found according to this order is the parameter that is used.

The method for specifying these installation and customization parameters is by using panel that is invoked by running an execution module in EXEC(ADB2CUST). When you run this execution module, the main installation customization menu (Figure 3-1) appears. It is from this panel that you can navigate to the general and subsystem-specific parameters panels. It is here that you also designate where you want your parameters to be saved. All of these parameters are saved to an ISPF table.

```
DB2 Admin ----- DB2 Administration - Customization ----- 12:03
Option ==>

Options:
  1 - General parameters for DB2 Admin
  2 - DB2 subsystem parameters for DB2 Admin
  3 - Copy general parameters and DB2 subsystem parameters

Library to be used for DB2 Admin customization ISPF tables:
ISPF table library ==> 'ADB.V72RLSR1.ISPTLIB'
```

Figure 3-1 DB2 Administration - Customization menu

If you select option 1 from the main customization menu, you see the general parameters panel (Figure 3-2). On this panel, various options can be chosen; their meanings are explained in the *IBM DB2 Administration Tool for z/OS User's Guide*, SC19-3033.

If you select option 2 from the main customization menu (Figure 3-1 on page 66), you see the DB2 subsystem parameters menu (Figure 3-3 on page 68). This panel shows the two options that can be chosen in addition to the active DB2 subsystems on this MVS™ subsystem, and displays the ISPF table library where changes are saved. The ISPF table library cannot be changed on this panel; you have to go back to the main customization menu to change the ISPF table library.

If you select option 3 from the main customization menu (Figure 3-1 on page 66), the customization parameters panel opens (Figure 3-4 on page 69).

```
DB2 Admin ----- Customization - General Parameters ----- 12:04
Option ==>

General parameters for DB2 Admin:

Press ENTER to save parameters, or END to leave without saving parameters.
More:      +

DB2 security exit type          ==> STD
(STD,SAMPLE,AUTH,NOCALL,OWN)
Copy version control table qualifier ==>          (table owner ID)
System identification method      ==>          (JESID,SMFID,SYSNAME,NONE)
Unit name for TSO work data sets  ==> SYSALLDA
Unit name for batch data sets     ==> SYSALLDA
Installation name                 ==>
Node Name                         ==>
JES3 system                       ==>          (Yes/No)
Unicode translation technique     ==> ER        ('ER      ', etc)
ISPF LLib1(not lnk1st'd) ==> SPF.PRODUCT.ISPLOAD
ISPF LLib2(not lnk1st'd) ==>
ISPF Message Library              ==> SPF.PRODUCT.ISPMLIB
ISPF Table Library                ==> SPF.PRODUCT.ISPTLIB
DB2 Admin APF Library             ==>

Current SYSAFF information for this MVS system:
JES ID:          NONE (found via JESID method)
SMF ID:          SY4A (found via SMFID method)
MVS system name: STPL (found via SYSNAME method)
```

Figure 3-2 Customization - General Parameters panel

On this panel, you can enter the desired information and press Enter to save the information to the ISPF table. The meanings of each field are described in Chapter 2, “Activating DB2 Admin”, in *IBM DB2 Administration Tool for z/OS User's Guide*, SC19-3033.

If you select option 1 on the DB2 subsystem parameters menu (see Figure 3-3), it adds one record with null values for each active DB2 subsystem that is not already present in the table. No information for DB2 subsystems already present in this table is altered. Upon successful addition of these records, a message is returned to the user.

```
DB2 Admin ----- Customization - DB2 Subsystem Parameters ----- 12:44
Option ==>

Options:
  1 - Merge list of active DB2 subsystem(s) into ISPF table
  2 - Edit ISPF table

Currently active DB2 subsystem(s) on this MVS system:
DB2 subsystem(s):   DSN6 DSN5 DSN7 DSN8
Data sets to be used for DB2 Admin customization:
ISPF table library: 'ADB.V72RLSR1.ISPTLIB'
```

Figure 3-3 Customization - DB2 Subsystem Parameters panel

If you select option 2 from the DB2 subsystem parameters menu (Figure 3-3), you receive the ISPF edit panel for the subsystem information (Example 3-1).

Example 3-1 Customization ISPF table edit

```
*
* Tags recognized by ADB2CUST EXEC:
* -----
* :nick.      Name of DB2 subsystem or DB2 group being described
* :desc.      Text to be displayed for this DB2 subsystem or DB2 group
* :secexit.   DB2 security exit type (STD,SAMPLE,AUTH,OWN,NOCALL)
* :cvtown.    Copy version table owner
* :stcnam.    Started task name of this DB2 subsystem
* :grpnam.    DB2 group name of this DB2 subsystem or DB2 group
* :jclass.    Job class to be used for batch DB2 utility jobs
* :jsysaff.   SYSAFF to be used for batch DB2 utility jobs
* :instparm.  Installation name
* :utilpre.   Utility data set prefix (USERID,OWNER,CREATEDBY,name)
* :ssid.      Subsystem name of the remote subsystem (blank if local)
* :nodenam.   Node name of the remote subsystem (blank if local)
* :locnam.    Location name of the remote subsystem (blank if local)
* :authsw.    Authorization Switching enabled (YES,NO)
* :newappl.   ISPF application ID
* :prompt.    Default value for Prompt Options (YES,NO)
* :resetopt.  Default value for Reset to Default at Startup (YES,NO)
* :nstuproc.  Number of job steps in the DSNUPROC procedure
* :cmown.     Owner (Qualifier) of Change Management database objects
* :swssid.    Allow switch of SSID
* :authswid.  DB2 Security ID to use for auth-switching
* :concent.   DB2 CONCENTRATE STATEMENTS WITH LITERALS (YES,NO)
* :currcom.   DB2 Use CONCURRENT clause on SQL (YES,NO)
* :hpu.       High Performance Unload (HPU) enabled (YES,NO)
* :hpulib.    High Performance Unload (HPU) load library
* :hpuplib.   High Performance Unload (HPU) parmlib library
* :uexelib.   REXX user exit library
```

From here you can specify parameters at two different levels. One is generic for any subsystem (using the nick.* header) and specifies those parameters that are global to all subsystems and DB2 subsystem-specific parameters. The parameter meanings are described in Chapter 2, “Activating DB2 Admin”, in *IBM DB2 Administration Tool for z/OS Users Guide*, SC19-3033. After making any and all modifications you want to make, you can press the PF key associated with the END command to return to the DB2 subsystem parameters menu.

Figure 3-4 Customization - Copy Parameters panel

Chapter 3. Product parameters **69**

3.3 General operational parameters

In this section, we discuss the parameters that manage behavior of the product at the user level. These parameters control everything from ISPF settings to display limits and job cards for generated jobs.

The main DB2 Administration Tool option panel (Figure 3-5) is invoked by selecting option P from the main DB2 Administration Menu. All of the parameters modified throughout these panels are stored as ISPF variables.

```
DB2 Admin ----- DB2 Change DB2 Admin Options ----- 12:54
Option ==>

    0 - Change ISPF settings                                DB2 System: DSNA
    1 - Change colors and highlights                        DB2 SQL ID: DBAUSER
    2 - Change DB2 Admin defaults
    A - Change alter options
    BP - Change batch parameters
    M - Change migrate options
    P - Change/allocate print data set
    PR - Change prompt options
```

Figure 3-5 DB2 Change DB2 Admin Options panel

If you select option 0 (Change ISPF Settings), the ISPF Settings menu opens. For an explanation of these parameters, refer to Chapter 2, “Settings (option 0)”, in *ISPF User's Guide Volume II*, SC34-4823.

If you select option 1 on the main DB2 Administration Tool option panel, you see the DB2 Administration Tool Change Colors and Highlights panel (Figure 3-6).

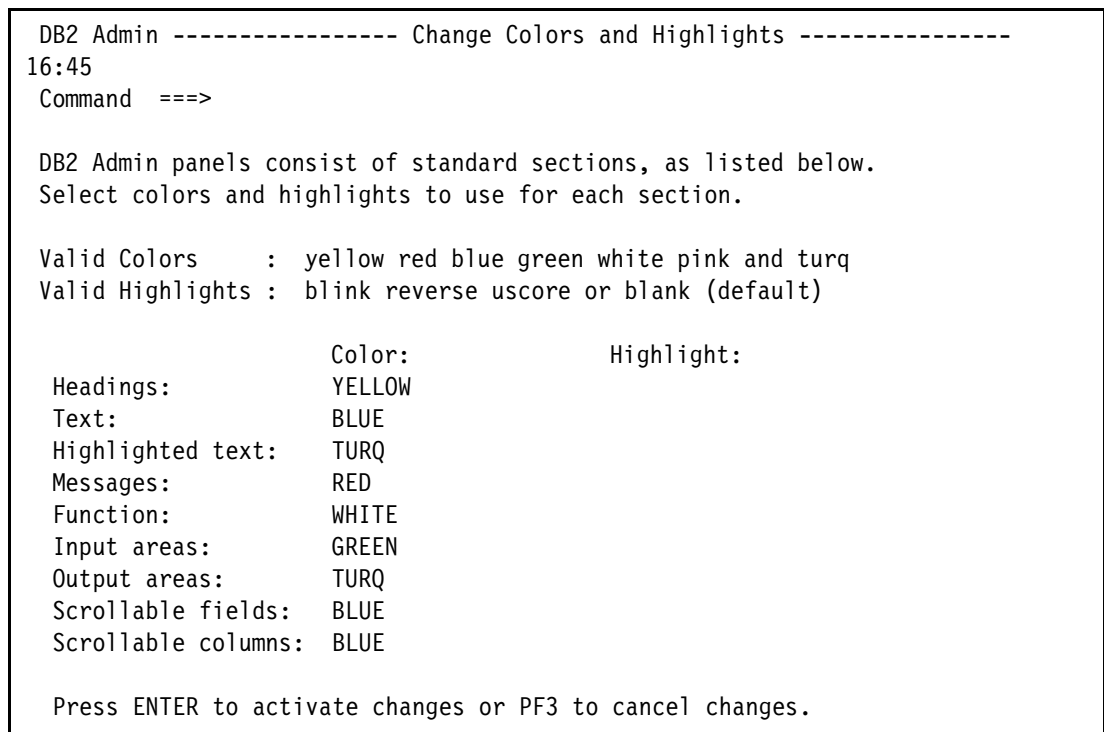


Figure 3-6 Change Colors and Highlights panel

Option 2 on the main DB2 Administration Tool option panel presents the user with the DB2 Admin default parameters panel (Example 3-2). This panel can also be invoked from various places in the product by entering the primary command of 'PARMS'.

Option A on this panel brings the user to the alter parameters panel (Figure 3-7 on page 74).

Option BP allows the user to change parameters pertaining to job control language (JCL) job cards, batch restart parameters (ADBTEP2), and data set space (seeFigure 3-8 on page 75).

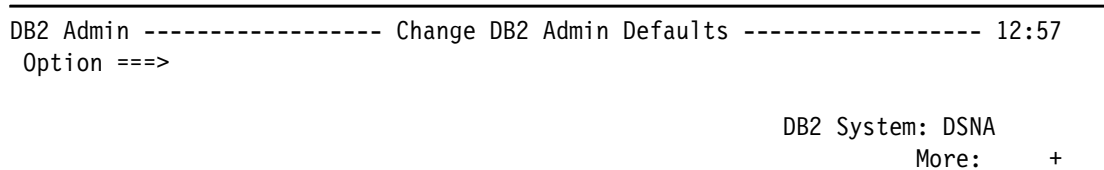
Option M brings the user to the migrate parameters panel (Figure 3-9 on page 76).

Option P defines the print data set.l

Option PR shows the user the prompt options panel (Figure 3-10 on page 76). You can also open this panel by entering the 'PROMPT' primary command on any other panel.

On the Change Colors and Highlights panel (Figure 3-6) you can control the color or the highlighting of various types of fields based on their type according to your preference by entering the colors and highlights desired and pressing Enter to save your changes.

Example 3-2 DB2 Administration Tool defaults



Max No of Rows to Fetch . . .	1000	(0-327670, 0=unlimited, def. 1000)
Max Chars in an SQL Stmt . . .	32765	(4000-32765, default is 32765)
Pgm Action when SQL error:		
First do a	ROLLBACK	(Commit or Rollback)
Display error panel	YES	(Yes/No)
Continue executing SQL . . .	NO	(Yes/No)
Auto Refresh After Update . .	YES	(Yes/No, default is YES)
Display SQL cost estimate . .	YES	(Yes/No, default is YES)
Browse DB2 Command Output . .	YES	(Yes/No)
Max Chars in an ISPF Stmt . .	2000	(500-32765, default is 2000)
Max Chars in an Admin Cmd . .	32765	(500-32765, default is 32765)
Report Drop Impacts	YES	(Yes/No)
Report Revoke Impacts	YES	(Yes/No)
Reset to Def. at Startup . . .	YES	(Yes/No)
Action when no rows found . .	M	(M - Message (default), P - Panel)
Default local CCSID	00000	(Optional, numeric)
Verify CCSID	YES	(Yes/No, default is YES)
Capitalize object names . . .	YES	(Yes/No, default is YES)
Capitalize data	YES	(Yes/No, default is YES)
Use trusted context in batch .	NO	(Yes/No, default is NO)
Gen. utilities for restricted	NO	(Yes/No, default is YES)
Display result of explain . .	NO	(Yes/No, default is NO)

From this panel, you can set parameters to your own preferences. These parameters are at the user level, so each user can set these parameters to suit their own tastes. For a detailed explanation of each of these parameters, refer to the *IBM DB2 Administration Tool for z/OS User's Guide*, SC19-3033.

Three parameters that deserve particular mention are “Max No of Rows to Fetch”, “Auto Refresh After Update”, and “Reset to Def. at Startup”.

“Max No of Rows to Fetch” limits the number of rows fetched from the DB2 catalog, but keep in mind that the result set shown on the panel may be incomplete. For example, if you have this parameter set to 1000 (which is the default) and you query all the tables in a subsystem, the number of rows fetched stops at 1000. If you then issue a primary command on this panel, the primary command operates only against those objects you have fetched. If a query reaches the user-specified maximum of rows fetched, a warning message is shown. A value of zero in this parameter denotes that nolimit should be used.

Note: This parameter is valuable in that it allows you to avoid unnecessary overhead when issuing queries that return a large number of objects, but you should make sure that the result sets you expect are complete by not setting this parameter to too small a value.

Alternatively, you can use the Resource Limit Facility (RLF) to warn or restrict runaway queries. DB2 Admin supports the SQL codes provided by RLF.

“Auto Refresh After Update” indicates that the tool should refresh the information shown on the panels if an update occurs. For example, supposed you were to show the index panel for a list of indexes and change the CLUSTER attribute for an index using the AL (alter) line command from (Y)es to (N)o. If the “Auto Refresh After Update” parameter was set to Yes, when you return to the index panel, the CLUSTER attribute reflects the change you made. If the “Auto Refresh After Update” parameter was set to No, when you return to the index panel, the CLUSTER attribute does not reflect this most recent change and you must manually refresh the panel by using the REFRESH primary command.

Note: Keeping the “Auto Refresh After Update” parameter set to Yes should not incur any significant impact to performance.

“Reset to Def. at Startup” upon installation is set to Yes for every user. This parameter resets the default values upon reentry into the product for several other parameters when it is set to No, which is the default setting. The parameters that are reset to their respective defaults are:

- ▶ Max No of Rows to Fetch
- ▶ Max Chars in an SQL Stmt
- ▶ Auto Refresh After Update
- ▶ Max Chars in an ISPF Stmt
- ▶ Max Chars in an Admin Cmd

Tip: Change this parameter to No to avoid unexpected changes to the other parameters upon reentry into DB2 Administration Tool.

The parameters shown in Figure 3-7 manage options that pertain to the ALT line command (see Chapter 7, “The ALT line command” on page 165 for more specific information about the ALT line command).

```

DB2 Admin ----- DSNA ALTER Analysis Options -----
10:38
Option ==>

Please specify the following for DB2 Admin ALTER:

Analysis options:
  Run SQLID . . . . . (Blank, an SQLID, or <NONE>)
  Use DEFER YES . . . . . NO (Yes/No)
  VIEW Column List . . . . . YES (Yes/No)
  Perform recovery analysis . . . . . NO (Yes/No)
  Enable authorization switching . . . NO (Yes/No)

Perform analysis in batch . . . . . YES (Yes/No)

Show this panel prior to each use . . . YES (Yes/No)

Change diagnostic options . . . . . NO (Yes/No)

```

Figure 3-7 DSNA ALTER Analysis Options panel

One parameter to mention is “Show this panel prior to each use”. If it is set to Yes, you see this panel whenever you perform the ALT line command.

Tip: If you find that you seldom change these options, you can change “Show this panel prior to each use” to No.

For a detailed explanation of the ALTER parameters, refer to *IBM DB2 Administration Tool for z/OS User's Guide*, SC19-3033 or 7.1.2, “ALT change process” on page 167.

In the panel shown in Figure 3-8, you can control information pertaining to the generation of job cards, jobparms, and control the batch restart program (ADBTEP2) defaults and information controlling the disposition of data sets based on various space parameters.

```

DB2 Admin ----- DSNA Batch Job Utility Parameters -----
15:47
Command ==>

Generate Job Card . . . (Yes/No)
Job cards:
====> //DBAUSERD JOB , 'DB2 UTILITY',
====> //          REGION=OM, NOTIFY=DBAUSER,
====> //          MSGCLASS=X,
====>
====>
Generate Job CLASS . . YES (Yes/No)    JOB CLASS . . . . .

JOBPARM:
====>
====>
====>
====>

ADBTEP2:
Restart . . . . . (Yes/No)
Maxerrors . . . . . (-1 to 99)
BindError . . . . . MAXE (MAXE, Save or Ignore)
Log DIAG . . . . . NO (Yes/No)
AutoCheck . . . . . NO (Yes/No)
LOAD Summary Report . YES (Yes/No)
Advisory Auto Rebuild. NO (Yes/No)
Advisory Auto Reorg . NO (Yes/No)
DB2 Pending Changes options:
  Check at DROP . . . YES (Yes/No)

Space parameters:
Unit name . . . . . SYSDA
Space unit . . . . . TRK (BLK, TRK, CYL or 4096-32760)
Max Primary . . . . . 65535 (In above units, 99999999 or blank)
                               In KB: 3145680
Max DASD . . . . . 65535 (In above units. Allocations beyond this
                               are sent to tape) In KB: 3145680
Tape Unit . . . . . TAPE (Unit for tape if size is greater
                               than Max DASD)

Default space allocation if unable to calculate:
Primary alloc . . . . 30 (In above units)
Secondary alloc . . . 30 (In above units)

```

Figure 3-8 DSNA Batch Job Utility Parameters panel

Note: When you specify that a job card should be generated and you also want to generate the job class, make sure to end the last entered job card with a comma, as shown in Figure 3-8. If this comma is left off, you are likely to receive JCL errors.

Further information about the ADBTEP2 parameters can be found in *IBM DB2 Administration Tool for z/OS User's Guide*, SC19-3033. For a full explanation of all of these parameters, refer to *IBM DB2 Administration Tool for z/OS User's Guide*, SC19-3033.

The only parameter on this panel (Figure 3-9) controls whether space information is gathered when the MIG primary command is invoked. The Migrate Table Spaces panel (ADB28S) has a column to show VSAM space for each page set in KB. If you enter this panel with this parameter set to No, you can always issue the SPACE primary command to show the space information after the panel is shown. If this parameter is set to Yes, it can take additional time to build the panel. It should only be a noticeable delay if a significant number of objects are shown on this panel.

```
DB2 Admin ----- Change Migrate Options -----
15:48
Option ==>

Show space information on panels . . YES (Yes/No, default is YES)
```

Figure 3-9 Change Migrate Options panel

For a full explanation of all of these parameters, refer to *IBM DB2 Administration Tool for z/OS User's Guide*, SC19-3033.

This panel (Figure 3-10) controls the types of statements that prompt you before the statements are executed. You can get to this panel by using the PR option from the DB2 Admin Options panel or enter PROMPT as a primary command on any panel. You can also turn all of these switches on or off using the primary commands PROMPT ON or PROMPT OFF on any panel, respectively.

```
DB2 Admin ----- Prompt Options -----
15:50
Option ==>

Change one or more options below. Prompt before executing:

Definition SQL (CREATE, DROP, ALTER, RENAME,.) ==> NO (Yes/No)

Authorization SQL (GRANT and REVOKE) ==> NO (Yes/No)

Update SQL (INSERT, UPDATE, DELETE) ==> NO (Yes/No)

DSN commands (BIND, REBIND and FREE) ==> NO (Yes/No)

DB2 commands (START, STOP, ALTER, SET) ==> NO (Yes/No)
```

Figure 3-10 Prompt Options panel

3.4 Change management ID parameters

In this section, we discuss the change management ID facility that is used to control the behavior of change management with regard to individual users. Both a default and user-specific behavior may be set on this panel.

The primary means to open the change management facility is to use the CM command on the main DB2 Administration Tool menu or use the CMM primary command on any panel. Upon entering either of these commands, the Change Management (CM) panel opens (Figure 3-11).

```
DB2 Admin ----- Change Management (CM) -----
10:11
Option ==>

      1 - Manage changes
      2 - Manage masks
      3 - Manage ignores
      4 - Manage versions
      5 - Manage ID table
      6 - Report changes

DB2 System: DSNA
DB2 SQL ID: DBAUSER
CM Owner   : ADB
```

Figure 3-11 Change Management (CM) panel

This panel is used to manage any and all change management information. This chapter only discusses option 5 (Manage ID table). For more information about the rest of these options, refer to the *IBM DB2 Administration Tool for z/OS User's Guide*, SC19-3033.

Selecting option 5 on this panel opens the CM - Manage ID Table panel (Figure 3-12).

```
DB2 Admin ----- CM - Manage ID Table ----- Row 1 to 12 of 22
Command ==>                                     Scroll ==> PAGE

Commands: CANCEL
Line commands: I - Insert D - Delete

Change management default level setting : NONE

Sel SQL ID   Level   Oper
  *         *       *
-----
JONESRJ  OPTIONAL
SMITHAB   REQUIRED
DBAUSER   OPTIONAL
PETERST   OPTIONAL
WONGJRT   OPTIONAL
```

Figure 3-12 CM - Manage ID Table panel

You are able to set the change management behavior both for a specific user and a default user. The default user is used if a specific user is not defined in this table. This information is stored in the change management database, which is defined as part of the installation of this product.

On this panel, you can set the three change management behavior settings. They are:

► **NONE**

This option disables change management for that user. No options to use change management appears for that user.

► **OPTIONAL**

This option presents the user with a menu in appropriate situations to allow the user to choose whether to use change management or not.

► **REQUIRED**

This option forces the user to use change management for all appropriate situations.

In the example shown in Figure 3-12 on page 77, the default behavior for all users except those specifically shown is to disable change management. For the JONESRJ, DBAUSER, PETERST, and WONGJRT users, you have a choice to use change management. The SMITHAB user is forced to use change management.



Part 3

DB2 9 for z/OS and prior support

In this part, we describe some of the DB2 9 for z/OS features that have been supplied by the DB2 Administration Tool V10 and prior releases of the product.

This part contains the following chapters:

- ▶ Chapter 4, “Native SQL procedures” on page 81
- ▶ Chapter 5, “Universal table space” on page 109
- ▶ Chapter 6, “CLONE tables” on page 143



Native SQL procedures

In this chapter, we describe what a native SQL stored procedure (NSP) is and how you can create and manage the native stored procedure, including bind deploy the stored procedure to a different location using the DB2 Administration Tool.

This chapter contains the following topics:

- ▶ What a native SQL stored procedure is
- ▶ Using the CREATE PROCEDURE function to create native SQL stored procedures
- ▶ Managing a native SQL stored procedure

4.1 What a native SQL stored procedure is

A native SQL stored procedure stores all its SQL statements and converts them to a native representation in the DB2 catalog and directory, as is done with other SQL statements. The parameter list and procedure are stored in the database catalog tables like other stored procedures. With DB2 9 for z/OS new function mode or higher, when you CALL a native SQL procedure, DB2 loads the native representation from the directory and executes the procedure.

You do not need a C or C++ compiler for native SQL stored procedures. The SQL statements run entirely within the DB2 engine. Conversely, external SQL procedures require a C compiler, exist as external load modules, and are executed in the WLM environment.

All SQL procedures that are created without the FENCED or the EXTERNAL option in the CREATE PROCEDURE statement are native SQL procedures. In previous releases of DB2, you did not specify either of these options; the procedures were created as external procedures.

Native SQL stored procedures offer enhanced support for the SQL Procedural Language, which implies new constructs such as FOR loops, nested compound statements, and more data types (for example, BIGINT, BINARY, VARBINARY, and DECFLOAT).

The maximum length of the native SQL procedure statement is 2 MB in total, including the CREATE PROCEDURE keyword, parameters, and the SQL body.

You can define multiple versions of the stored procedure. CREATE PROCEDURE is used to define the initial version, and ALTER PROCEDURE is used to define subsequent versions.

Using the DB2 Administration Tool, you can create a version of a SQL stored procedure, debug it, replace it, or add a new version of the procedure, and finally deploy it.

4.2 Using the CREATE PROCEDURE function to create native SQL stored procedures

To be able to use the CREATE PROCEDURE SQL statement, you must have at least one of the following authorizations:

- ▶ The CREATEIN privilege on the schema that you are using
- ▶ SYSADM or SYSCTRL authority

The DB2 Admin supports a CREATE PROCEDURE option where you may define a native stored procedure. The option provides editing capabilities of the CREATE PROCEDURE statement after the stored procedure in/out parameters and stored procedure bind options have been entered. The options open an edit session so you can write the SQL body of the stored procedure.

We now show a step by step example of creating a native SQL procedure using the Create Procedure function of DB2 Admin:

1. To create a native stored procedure using DB2 Admin Tool, enter 2.4 on the option line of the main menu (ADB2) and press Enter (Figure 4-1).

```
ADB2 dmin ----- DB2 Administration Menu 10.1.0 ----- Option ==>
2.4

1 - DB2 system catalog                      DB2 System: VA1A
2 - Execute SQL statements                  DB2 SQL ID: ADMR2
3 - DB2 performance queries                Userid   : SYSADM
4 - Change current SQL ID                  DB2 Schema: ADMR2
5 - Utility generation using LISTDEFS and TEMPLATES DB2 Rel   : 1015
P - Change DB2 Admin parameters
DD - Distributed DB2 systems
E - Explain
Z - DB2 system administration
SM - Space management functions
W - Manage work statement lists
X - Exit DB2 Admin
CC - DB2 catalog copy version maintenance
CM - Change management

More:      +

Interface to other DB2 products and offerings:
I  DB2I

C  DB2 Object Comparison Tool
```

Figure 4-1 Option 2.4 from DB2 Administration Menu

2. Enter C0 - Create Stored Procedure on the option line of the Create/Drop/Label/Comment On Objects panel (ADB26COU) (Figure 4-2).

After you hit Enter, the Create Procedure panel (ADB26CO) opens. Enter the information of the SQL procedure that you are about to create.

```

ADB26 min ----- VA1A Create/Drop/Label/Comment On Objects ----- Option ==>
C0

                                     More:      +
                                     DB2 System: VA1A
                                     DB2 SQL ID: ADMR2

CREATE                                DROP
CG - Storage group                    DG - Storage group
CD - Database                         DD - Database
CS - Table space                      DS - Table space
CT - Table                            DT - Table
CV - View                             DV - View
CL - Alias                            DL - Alias
CX - Index                            DX - Index
CY - Synonym                          DY - Synonym
CA - Auxiliary table
CE - Distinct type                    DE - Distinct type
CJ - Trigger                          DJ - Trigger
CF - Function                          DF - Function
CO - Stored procedure                 DO - Stored procedure
CM - Materialized table                DQ - Sequence
CQ - Sequence
CTR - Trusted context                  DTR - Trusted context
CRO - Role                             DRO - Role
LABEL                                COMMENT (remark)

```

Figure 4-2 Specify CO (Create Stored Procedure) option on ADB26

4.2.1 Create Procedure panel (ADB26CO)

On the Create Procedure panel (ADB26CO), you need to enter the following information for the stored procedure:

Schema	The schema of the stored procedure. Schema is a logical grouping of SQL objects. It is used as a qualifier for the name of the stored procedure.
Name	The name of the stored procedure.
Number of parameters	The number of parameters of the procedure.
Language	The language in which the procedure is written.
Native SP	Indicates whether this is a native SQL procedure.
Version	The version identifier of the native SQL procedure to be generated. The identifier can be up to V64 EBCDIC bytes. V1 is the default identifier if the field is left blank.

If you specified that 0 for the Number of parameters for the stored procedure, the CREATE Stored Procedure Parameters panel (ADB26COU) is skipped and the Create Stored Procedure Option panel (ADB26COV) opens.

In our example, we create a SQL native stored procedure named DMR2.NSP_FIND_MEMID that has three parameters (two INPUT parms and one OUTPUT parm) and its first version identifier is V10142010 (Figure 4-3).

```

ADB26CO n ----- VA1A Create Procedure ----- 13:56
Command ==>

CREATE PROCEDURE

Schema . . . . . ADMR2      >      (Default is ADMR2)
Name . . . . . NSP_FIND_MEMID    >  (? to look up existing procedures)

(
  Number of parameters . . 3      (0-255)
)

LANGUAGE . . . . SQL          (ASSEMBLE,C,PLI,COBOL,REXX,JAVA,SQL)

Native SP . . . Y              (Yes,No)
VERSION . . . V10142010        >  (optional, default is V1)

```

Figure 4-3 Create SQL native stored procedure

After you press Enter, the CREATE Stored Procedure Parameters panel (ADB26COU) opens.

4.2.2 CREATE Stored Procedure Parameters (ADB26COU)

The CREATE Stored Procedure Parameters panel (ADB26COU) is used to define the stored procedure parameters. At a minimum, you need to specify the following:

Parm type	The usage of the parameter. The valid values are IN, OUT, and INOUT.
IN	Identifies the parameter as an input parameter to the procedure. IN is the default.
OUT	Identifies the parameter as an output parameter that is returned by the procedure. If the parameter is not set within the procedure, the null value is returned.
INOUT	Identifies the parameter as both an input and output parameter for the procedure. If the parameter is not set within the procedure, its input values is returned.
Parm name	The parameter name.
Data type	The data type of the parameter. The data type of the parameter is a built-in data type.
Length	The length of the data.
Scale	The scale for DECIMAL or TIMESTAMP data type.

In our case, we have specified three parameters for the stored procedure.

- ▶ The first one is an input parameter, V_MEM_NAME, with data type CHAR of length 15.
- ▶ The second one is an input parameter, V_MEM_ADDR, with data type CHAR of length 45.

- The third one is an output parameter, V_MEM_ID, with data type INTEGER.

The ADB26COU is presented three times, once for each parameter to be defined (Figure 4-4, Figure 4-5 on page 87, and Figure 4-6 on page 87).

```

ADB26COU ----- VA1A CREATE Stored Procedure Parameters ----- 13:5
Command ==>

                                                                    More:  +
CREATE PROCEDURE "ADMR2"."NSP_FIND_MEMID"
  (parameter number 1)          LANGUAGE SQL

Parm type  . . . IN              (IN, OUT, or INOUT)
Parm name  . . . V_MEM_NAME      >  (Parameter name)

For a non table like parameter specify:

Data type  . . . CHAR            >  (Built-in only)
Length     . . . . . 15          (1 if DBCLOB with units indicator G)
Scale      . . . . .

FOR ? DATA . . .                (BIT, SBCS, or MIXED)
CCSID      . . . . .            (ASCII, EBCDIC, or UNICODE)
AS LOCATOR . . .                (Yes/No)

For a TABLE LIKE parameter specify:
  Table owner . .              >  (Default is SYSADM)
  Table name  . . .           >  (Table parameter, ? to look up)
  AS LOCATOR

```

Figure 4-4 Enter values for Parameter 1

```
ADB26COU ----- VA1A CREATE Stored Procedure Parameters ----- 13:58
Command ==>
```

More: +

```
CREATE PROCEDURE "ADMR2"."NSP_FIND_MEMID"
(parameter number 2)      LANGUAGE SQL
```

```
Parm type . . . IN          (IN, OUT, or INOUT)
Parm name . . . V_MEM_ADDR  > (Parameter name)
```

For a non table like parameter specify:

```
Data type . . . CHAR          > (Built-in only)
Length . . . . . 45          (1 if DBCLOB with units indicator G)
Scale . . . . .
```

```
FOR ? DATA . . .          (BIT, SBCS, or MIXED)
CCSID . . . . .            (ASCII, EBCDIC, or UNICODE)
AS LOCATOR . . .          (Yes/No)
```

For a TABLE LIKE parameter specify:

```
Table owner . .          > (Default is SYSADM)
Table name . . .         > (Table parameter, ? to look up)
AS LOCATOR
```

Figure 4-5 Enter value for Parameter 2

```
ADB26COU ----- VA1A CREATE Stored Procedure Parameters ----- 13:58
Command ==>
```

More: +

```
CREATE PROCEDURE "ADMR2"."NSP_FIND_MEMID"
(parameter number 3)      LANGUAGE SQL
```

```
Parm type . . . OUT          (IN, OUT, or INOUT)
Parm name . . . V_MEM_ID    > (Parameter name)
```

For a non table like parameter specify:

```
Data type . . . INTEGER      > (Built-in only)
Length . . . . .            (1 if DBCLOB with units indicator G)
Scale . . . . .
```

```
FOR ? DATA . . .          (BIT, SBCS, or MIXED)
CCSID . . . . .            (ASCII, EBCDIC, or UNICODE)
AS LOCATOR . . .          (Yes/No)
```

For a TABLE LIKE parameter specify:

```
Table owner . .          > (Default is SYSADM)
Table name . . .         > (Table parameter, ? to look up)
AS LOCATOR
```

Figure 4-6 Enter values for parameter 3

After all of the stored procedure parameters have been defined, you need to complete the CREATE Stored Procedure options panel. The options determine how the stored procedure executes.

4.2.3 CREATE Stored Procedure options (ADB26COV)

In the case of the native SQL stored procedure, DEBUG MODE is a new option. It specifies whether this version of the procedure can be run in debugging mode. The default is determined using the value of the CURRENT DEBUG MODE special register. The valid options for DEBUG MODE are as follows:

ALLOW	Specifies that this version of the procedure can be run in debugging mode. When this version of the procedure is invoked and debugging is attempted, a WLM environment must be available.
DISALLOW	Specifies that this version of the procedure cannot be run in debugging mode. You can use an ALTER PROCEDURE statement to change this option to ALLOW DEBUG MODE for this version of the procedure.
DISABLE	Specifies that this version of the procedure can never be run in debugging mode. This version of the procedure cannot be changed to ALLOW DEBUG MODE or DISALLOW MODE. To change this option, you need to drop the procedure and create it again using the desired option. An alternative is to create a version of the procedure using the desired option and make that version the active version.

In our example, we specified ALLOW DEBUG MODE, so a WLM environment must exist. You need to know the WLM application environment name and specify the WLM environment used by DB2 when debugging the procedure (Figure 4-7).

ADB26COV ----- VA1A CREATE Stored Procedure Options ----- 14:04

Command ==>

More: +

CREATE PROCEDURE "ADMR2"."NSP_FIND_MEMID"

(IN "V_MEM_NAME" CHAR(15) , IN "V_MEM_ADDR" CHAR(45) , OUT "V_MEM_ID" >

LANGUAGE SQL

PARAMETER CCSID . . (ASCII, EBCDIC, or UNICODE)

RESULT SETS (Maximum number of result sets. 0-32767)

DETERMINISTIC . . . (Yes, No)

CALLED ON NULL . . . (Yes)

SQL DATA (C - Contain, R - Read, M - Mod)

DEBUG WLM ENVIR . . **WLMENV1** > (Debug WLM environment name)

DEBUG MODE **ALLOW** (Disallow, Allow, Disable)

ASUTIME LIMIT . . . (CPU service units or 0 for no limit)

SPECIAL REGISTERS . (I - Inherit, D - Default)

COMMIT ON RETURN . . (Yes, No)

Figure 4-7 Specify Stored Procedure Options

The native stored procedure is bound in a package. After you press Enter, you are prompted to provide the BIND options for the package.

4.2.4 Create Stored Procedure BIND Options (ADB26COW)

On this panel, you specify the bind option for the stored procedure package. The default values are used if you press Enter without providing any input, as in this example (Figure 4-8).

```
ADB26COW ----- VA1A Create Stored Procedure BIND Options ----- 14:05
Command ==>

                                                                    More:      +
CREATE PROCEDURE "ADMR2"."NSP_FIND_MEMID"
  ( IN "V_MEM_NAME" CHAR(15)    , IN "V_MEM_ADDR" CHAR(45)    , OUT "V_MEM_ID" >
  VERSION V10I02010 LANGUAGE SQL WLM ENVIRONMENT FOR DEBUG MODE WLMENV1 A >

PACKAGE OWNER . . . . . >
QUALIFIER . . . . . >
DEFER PREPARE . . . . . (Yes, No)
CURRENT DATA . . . . . (Yes, No)
DEGREE . . . . . (1, Any)
DYNAMICRULES . . . . . (Run, Bind, DefineBind, DefineRun,
                        InvokeBind, InvokeRun)
APP ENCODING SCHEME . (ASCII, EBCDIC, UNICODE)
EXPLAIN . . . . . (Yes, No)
IMMEDIATE WRITE . . . (Yes, No)
ISOLATION LEVEL . . . (CS, RS, RR, UR)
KEEP DYNAMIC . . . . . (Yes, No)
OPT HINT . . . . . >
RELEASE . . . . . (Commit, Deallocate)
REOPT . . . . . (None, Always, Once)
VALIDATE . . . . . (Run, Bind)
DATE FORMAT . . . . . (ISO, EUR, USA, JIS, Local)
TIME FORMAT . . . . . (ISO, EUR, USA, JIS, Local)
DECIMAL PRECISION . . (15, 31)
  SCALE . . . . . (1-9)
DECIMAL ROUNDING . . . (Ceiling, Down, Floor, HalfDown,
                        HalfEven, HalfUp, Up)
FOR UPDATE CLAUSE . . (Required, Optional)
SQL PATH . . . . .
```

Figure 4-8 Enter Create Stored Procedure BIND Options

After you press Enter, an edit session opens in which you enter the body of the SQL statements of the SQL stored procedure.

4.2.5 Create SQL Stored Procedure Body (ADB26COQ)

In our panel, we entered the SQL statements of the native SQL stored procedure, NSP_FIND_MEMID (Figure 4-9).

```
ADB26COQ  ----- VA1A Create SQL Stored Procedure Body ---- Columns 00001 00072
Command ==>                                         Scroll ==> CSR

CREATE PROCEDURE "ADMR2"."NSP_FIND_MEMID"
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000100 BEGIN
000200 DECLARE NOR INTEGER ;
000300 DECLARE CSR11 CURSOR WITH RETURN FOR
000400 SELECT MEM_ID FROM ADMR2.MEMBER_LIST
000500 WHERE MEM_NAME = V_MEM_NAME AND MEM_ADDR = V_MEM_ADDR;
000600 OPEN CSR11 ;
000700 GET DIAGNOSTICS NOR = DB2_NUMBER_ROWS ;
000800 IF NOR < 1
000900 THEN
001000 SET V_MEM_ID = 11111111 ;
001100 END IF ;
001200 END
***** ***** Bottom of Data *****
```

Figure 4-9 Enter SQL stored procedure body

After keying in the SQL statements, press Enter and then press F3. The SQL stored procedure is created, as indicated by the CREATE stmt executed message on the Create Procedure panel (ADB26CO) (Figure 4-10). The SQL warn (+204) message is displayed if the object referenced in the SQL body, such as table ADMR2.MEMBER_LIST, does not already exist.

```
ADB26CO n ----- VA1A Create Procedure ----- 14:19
Command ==>
CREATE stmt executed
CREATE PROCEDURE

Schema . . . . . ADMR2      >      (Default is ADMR2)
Name . . . . . NSP_FIND_MEMID  >  (? to look up existing procedures)

(
Number of parameters . . 3      (0-255)
)

LANGUAGE . . . . . SQL      (ASSEMBLE,C,PLI,COBOL,REXX,JAVA,SQL)

Native SP . . . YES      (Yes,No)
VERSION . . . V10142010  >  (optional, default is V1)
```

Figure 4-10 SQL stored procedure created

You have successfully created a native SQL stored procedure.

You can specify up to 64 EBCDIC bytes version IDs. Only eight characters are displayed on the panel. You need to place the cursor under the > character and press PF11 to see the remaining ID (Figure 4-11).

```

ADB210 in ----- VA1A Stored Procedures ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Commands: GRANT
Line commands:
AH - Schema Auth A - Auth DROP - Drop AL - Alter K - Package PA - Parms
DIS - Display STO - Stop STA - Start GR - Grant COM - Comment
? - Show all line commands

                                S
                                Res Q S P C External
Sel Schema Name Version A Lang Parms Set 0 L R T R Name
    *      *      *      * *
----->-----
      ADMR2   NSP_FIND_MEMID  V1014201 Y SQL      3  0 N M N  N
***** END OF DB2 DATA *****

```

Figure 4-11 New native stored procedure created

4.3 Managing a native SQL stored procedure

Now that you have created a native SQL stored procedure, the following DB2 Administration Tool's line commands can help you managing the native stored procedure:

ADDV	Add version.
ACT	Activate version.
REG	Regenerate version.
DEPV	Replace version.
DROPV	Drop version.
BIND	Bind deploy version.

Because most of these line commands are self explanatory, we show examples for the ADDV, ACT, and BIND commands.

4.3.1 ADDV (ADD version) line command

The first version of the stored procedure created is, by default, the *active* version. You can ALTER the stored procedure and add additional versions.

The ADDV line command helps you add additional versions to the existing native SQL stored procedure. You can change the parameter names, procedure options, and procedure routine body for the new version of the native stored procedure.

Figure 4-12 shows an example of the ADDV line command.

```
ADB210 in ----- VA1A Stored Procedures ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT
Line commands:
AH - Schema Auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment
? - Show all line commands

                                     S
                                     Res  Q S P C External
Sel  Schema  Name                Version  A Lang Parms  Set  O L R T R Name
   *      *
----->-----
addv ADMR2    NSP_FIND_MEMID      V1014201 Y SQL        3   0 N M N  N
*****
***** END OF DB2 DATA *****
```

Figure 4-12 ADDV line command

After issuing the ADDV line command, the Add Procedure Panel (ADB26CO) opens, where you can enter the modified parameter related information for the new version you are about to add.

4.3.2 Add Procedure (ADB26CO)

On the Add Procedure panel, you need to specify the new version identifier (Figure 4-13).

```
ADB26CO n ----- VA1A Add Procedure ----- 14:40
Command ==>

ALTER PROCEDURE

Schema . . . . . ADMR2    >      (Default is SYSADM)
Name . . . . . NSP_FIND_MEMID    >  (? to look up existing procedures)

(
Number of parameters . . 3      (0-255)
)

ADD

VERSION . . . V10152010    >
```

Figure 4-13 Alter Procedure panel

Note: All versions of the stored procedure must have the same number of parameters.

After pressing Enter, the Modify Stored Procedure Parameters panel (ADB26COP) opens. This is an informational panel that informs that there are two options to proceed. You can modify the parameters of the stored procedure by pressing Enter and the Create Stored Procedure Parameters panel opens, where you can modify the parameters. You can skip the parameter panel by pressing End (Figure 4-14).

```
ADB26COP  -- VA1A Modify Stored Procedure Parameters ----- 14:44

You may choose whether to change the parameters.
Number of parameters: 3

Press ENTER to modify the parameters. This will invoke a series of
panels to allow you to modify them.

Press END to skip changes to the parameters. This will still result
in generation of a matching parameter list, but the parameter entry
panel is not invoked.
```

Figure 4-14 Modify stored procedure informational panel

In our example, we press F3 to skip changing the parameters. The Add Stored Procedure Options panel (ADB26COV) opens, where you can provide the procedure option change (Figure 4-15).

```
ADB26COV  ----- VA1A Add Stored Procedure Options ----- 14:50
Command ==>

More:      +
ALTER PROCEDURE "ADMR2"."NSP_FIND_MEMID" ADD VERSION V10152010
  ( IN "V_MEM_NAME" CHAR(15) FOR SBCS DATA  , IN "V_MEM_ADDR" CHAR(45) FOR >
LANGUAGE SQL

PARAMETER CCSID . . . . . (ASCII, EBCDIC, or UNICODE)

RESULT SETS . . . . . 0 (Maximum number of result sets. 0-32767)
DETERMINISTIC . . . . . NO (Yes, No)
CALLED ON NULL . . . . . YES (Yes)
SQL DATA . . . . . M (C - Contain, R - Read, M - Mod )

DEBUG WLM ENVIR . . WLMENV1 > (Debug WLM environment name)
DEBUG MODE . . . . . ALLOW (Disallow, Allow, Disable)
ASUTIME LIMIT . . . . . 0 (CPU service units or 0 for no limit)
SPECIAL REGISTERS . . I (I - Inherit, D - Default)
COMMIT ON RETURN . . NO (Yes, No)
```

Figure 4-15 Alter stored procedure options for ADDV

Press Enter if you do not want to change the stored procedure execute option, as in our example.

Next, the Add Stored Procedure BIND Options Panel (ADB26COW) opens, where you can change the BIND options (Figure 4-16).

ADB26COW ----- VA1A Add Stored Procedure BIND Options ----- 14:52	
Command ==>	
	More: +
ALTER PROCEDURE "ADMR2"."NSP_FIND_MEMID" ADD VERSION V10152010	
(IN "V_MEM_NAME" CHAR(15) FOR SBCS DATA , IN "V_MEM_ADDR" CHAR(45) FOR	>
LANGUAGE SQL RESULT SETS 0 NOT DETERMINISTIC CALLED ON NULL INPUT MODI	>
PACKAGE OWNER	SYSADM >
QUALIFIER	SYSADM >
DEFER PREPARE	(Yes, No)
CURRENT DATA	NO (Yes, No)
DEGREE	1 (1, Any)
DYNAMICRULES	RUN (Run, Bind, DefineBind, DefineRun, InvokeBind, InvokeRun)
APP ENCODING SCHEME .	(ASCII, EBCDIC, UNICODE) - CCSID = 37
EXPLAIN	NO (Yes, No)
IMMEDIATE WRITE . . .	NO (Yes, No)
ISOLATION LEVEL . . .	CS (CS, RS, RR, UR)
KEEP DYNAMIC	NO (Yes, No)
OPT HINT	>
RELEASE	COMMIT (Commit, Deallocate)
REOPT	NONE (None, Always, Once)
VALIDATE	RUN (Run, Bind)
DATE FORMAT	(ISO, EUR, USA, JIS, Local)
TIME FORMAT	(ISO, EUR, USA, JIS, Local)
DECIMAL PRECISION . .	15 (15, 31)
SCALE	(1-9)
DECIMAL ROUNDING . . .	HALFEVEN (Ceiling, Down, Floor, HalfDown, HalfEven, HalfUp, Up)
FOR UPDATE CLAUSE . .	(Required, Optional)
SQL PATH	

Figure 4-16 Alter stored procedure BIND options for ADDV

Press Enter if you do not want to change the stored procedure BIND option, as in our example. The Add SQL Stored Procedure Body (ADB26COQ) panel opens and shows the SQL body of the native stored procedure in an edit session so you can make modifications (Figure 4-17).

```
ADB26COQ  ----- VA1A Add SQL Stored Procedure Body ----- Columns 00001 00072
Command ==>                                         Scroll ==> CSR

ALTER PROCEDURE "ADMR2"."NSP_FIND_MEMID" ADD VERSION V10152010
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 BEGIN
000002   DECLARE NOR INTEGER ;
000003   DECLARE CSR11 CURSOR WITH RETURN FOR
000004   SELECT  MEM_ID FROM ADMR2.MEMBER_LIST WHERE MEM_NAME =
000005   V_MEM_NAME AND MEM_ADDR = V_MEM_ADDR ;
000006   OPEN CSR11 ;
000007   GET DIAGNOSTICS NOR = DB2_NUMBER_ROWS ;
000008   IF NOR < 1 THEN
000009       SET V_MEM_ID = 11111111 ;
000010   END IF ;
000011 END
***** ***** Bottom of Data *****
```

Figure 4-17 The SQL body of the stored procedure to be changed

In our example, we modified the SQL body, as shown in Figure 4-18.

```
ADB26COQ  ----- VA1A Add SQL Stored Procedure Body ----- Columns 00001 00072
Command ==>                                         Scroll ==> CSR

ALTER PROCEDURE "ADMR2"."NSP_FIND_MEMID" ADD VERSION V10152010
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 BEGIN
000002   DECLARE NOR INTEGER ;
000003   DECLARE CSR11 CURSOR WITH RETURN FOR
000004   SELECT  MEM_ID FROM ADMR2.MEMBER_LIST WHERE MEM_NAME =
000005   V_MEM_NAME AND MEM_ADDR = V_MEM_ADDR ;
000006   OPEN CSR11 ;
000007   GET DIAGNOSTICS NOR = DB2_NUMBER_ROWS ;
000008   IF NOR < 1 THEN
000009       SET V_MEM_ID = 99999999 ;
000010   END IF ;
000011 END
***** ***** Bottom of Data *****
```

Figure 4-18 Modified SQL body

After pressing F3, the Add Procedure panel (ADB26CO) opens with the ALTER stmt executed message under the Command line at the top of the panel. It is an indicator that the ALTER PROCEDURE ADD VERSION statement executed and the new version is successfully added (Figure 4-19).

```
ADB26CO n ----- VA1A Add Procedure ----- 15:06
Command ==>
ALTER stmt executed
ALTER PROCEDURE

Schema . . . . . ADMR2      >      (Default is SYSADM)
Name . . . . . NSP_FIND_MEMID    >  (? to look up existing procedures)

(
Number of parameters . . 3      (0-255)
)

ADD

VERSION . . . V10152010      >
```

Figure 4-19 Alter stored procedure ADD VERSION executed

Press F3 to return to the Stored Procedure panel (ADB21O), where there are two versions of the stored procedure displayed. The “Y” under column A indicates the one that is active. In our example, V10142010 is the Active version of the stored procedure (Figure 4-20).

```
ADB21O in ----- VA1A Stored Procedures ----- Row 1 to 2 of 2
Command ==>                                     Scroll ==> CSR

Commands: GRANT
Line commands:
AH - Schema Auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment
? - Show all line commands

                                     S
                                     Res  Q S P C External
Sel  Schema  Name                Version  A Lang Parms  Set 0 L R T R Name
      *      *                  *      * *      *  * * * * *
----->-----
      ADMR2   NSP_FIND_MEMID    V1014201 Y SQL      3   0 N M N   N
      ADMR2   NSP_FIND_MEMID    V1015201 N SQL      3   0 N M N   N
***** END OF DB2 DATA *****
```

Figure 4-20 ADB21O after ADDV executed successfully

ACT (ACTivate version) line command

You can have more than one version of the stored procedure. However, there is only one that is currently active.

To make the second version active, use the ACT line command and activate the specific version. Issue the ACT line command against version V10152010 and press Enter. The V10152010 now becomes the active version (Figure 4-21 and Figure 4-22).

```
ADB210 in ----- VA1A Stored Procedures ----- Row 1 to 2 of 2
Command ==>                                     Scroll ==> CSR

Commands: GRANT
Line commands:
AH - Schema Auth A - Auth DROP - Drop AL - Alter K - Package PA - Parms
DIS - Display STO - Stop STA - Start GR - Grant COM - Comment
? - Show all line commands

Sel Schema Name Version A Lang Parms Res Q S P C External
* * * * *
----->-----
ADM2 NSP_FIND_MEMID V1014201 Y SQL 3 0 N M N N
ACT ADM2 NSP_FIND_MEMID V1015201 N SQL 3 0 N M N N
***** END OF DB2 DATA *****
```

Figure 4-21 Activate version

```
ADB210 in ----- VA1A Stored Procedures ----- Row 1 to 2 of 2
Command ==>                                     Scroll ==> CSR
ALTER stmt executed
Commands: GRANT
Line commands:
AH - Schema Auth A - Auth DROP - Drop AL - Alter K - Package PA - Parms
DIS - Display STO - Stop STA - Start GR - Grant COM - Comment
? - Show all line commands

Sel Schema Name Version A Lang Parms Res Q S P C External
* * * * *
----->-----
ADM2 NSP_FIND_MEMID V1014201 N SQL 3 0 N M N N
ADM2 NSP_FIND_MEMID V1015201 Y SQL 3 0 N M N N
***** END OF DB2 DATA *****
```

Figure 4-22 Newly activated version

4.3.3 DRPV versus DROP line command

There are two types of DROP line commands that you can issue on the Stored Procedures panel (ADB21O):

- DROP** The DROP PROCEDURE statement is executed. *All* versions of the stored procedure all dropped.
- DRPV** The ALTER PROCEDURE DROP VERSION statement is executed and the specific version of the stored procedure is dropped. The identified version must *not* be the currently active version of the stored procedure. The SQLCODE -20315 is displayed if you try to drop the active version. You should use DROP line command instead of the DRPV line command to drop the stored procedure if there is only one version for the stored procedure and you want to drop the version.

You can deploy a native SQL procedure to the target DB2 You specify BIND PACKAGE DEPLOY COPYVER only when the target DB2 is a DB2 z/OS server. The minimum options that you need to specify for BIND DEPLOY are the target location, collection-ID, QUALIFER, ACTION, and OWNER. The default is used for the field that left blank.

If you specify ACTION(ADD) for a version that does not exist at the target location, DB2 creates or adds a new version of the native SQL procedure and its associated package.and keeps the procedure's SQL logic. DB2 adds a new version of a native SQL procedure with the same name if the named version already exists at the target location.

If you specify ACTION(REPLACE), DB2 DB2 replaces the version specified in COPYVER and creates a version if it does not exists at the target location.

4.3.4 BIND (BIND deploy version) line command

By issuing the BIND line command, you can deploy, ADD, or REPLACE the stored procedure at the specified location (Figure 4-23).

```
ADB21O in ----- VA1A Stored Procedures ----- Row 1 to 2 of 2
Command ==>                                     Scroll ==> CSR

Commands: GRANT
Line commands:
AH - Schema Auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment
? - Show all line commands
```

							S					External
Se1	Schema	Name	Version	A	Lang	Parms	Res	Q	S	P	C	
	*	*	*	*	*	*	Set	O	L	R	T	Name
----->-----												
	ADMR2	NSP_FIND_MEMID	V1014201	N	SQL	3	0	N	M	N	N	
BIND	ADMR2	NSP_FIND_MEMID	V1015201	Y	SQL	3	0	N	M	N	N	
***** END OF DB2 DATA *****												

Figure 4-23 BIND deploy stored procedure

You specify the name, collection ID, and qualifier all bind package related options, including the location where you want the stored procedure to be bound. In the following example, the stored procedure ADMR2.NSP_FIND_MEMID of the version V10152010 package will be bound and deployed to location STLEC3B (Figure 4-24).

With the ACTION(REPLACE), the stored procedure with the specific version is added if it does not exists if you use REPLACE.

```
ADB21KB n ----- VA1A Bind SQL Procedure Package -- --
Command ==>

More:      +

Verify BIND parameters:

BIND PACKAGE(
  Location . . . . . STLEC3B      >
  Collection . . . . . ADMR2      >
  OWNER . . . . . SYSADM      >
  QUALIFIER . . . . . SYSADM      >
  DEPLOY - collection ADMR2      >
  Deploy package . . . NSP_FIND_MEMID      >
  COPYVER . . . . . V10152010      >
  SQLERROR . . . . . NO      (COninue, NOpackage or CCheck)
  VALIDATE . . . . . R      (Run or Bind, Bind preferred)
  ISOLATION . . . . . CS      (CS, RR, RS, or UR)
  RELEASE . . . . . C      (Commit, Deallocate, or blank)
  EXPLAIN . . . . . NO      (Yes, No, or Only)
  CURRENTDATA . . . . NO      (Yes/No) (inhibit blocking)
  ACTION . . . . . REPLACE      (Add or Replace)
  REPLVER . . . . .
                                (replace version)
  ENABLE . . . . .
                                (use ? to get current values from the catalog)
```

Figure 4-24 Specify BIND deploy options

Press Enter and the BIND deploy executes successfully at the STLEC3B location (Figure 4-25).

```
ADB2CB in ----- VA1A Command Output Display -----
Command ==>                                Scroll ==> CSR

  BIND PACKAGE(STLEC3B.ADMR2) QUAL(SYSADM) OWNER(SYSADM) DEPLOY(ADMR2.NSP_FIND
  _MEMID) COPYVER(V10152010) ACTION(REPLACE)

***** Top of Data *****
DSNX100I > BIND SQL WARNING USING SYSADM AUTHORITY PLAN=(NOT APPLICABLE) DBR
DSNT232I @ SUCCESSFUL BIND FOR
          PACKAGE = STLEC3B.ADMR2.NSP_FIND_MEMID.(V10152010)
***** Bottom of Data *****
```

Figure 4-25 BIND deploy executed successfully

Now let us switch over to the DB2 subsystem at the STLEC3B location and verify that the stored procedure package is successfully deployed. The SSID primary command is used to switch over to the other system (refer to 11.1.2, “SSID” on page 284 for more information about this command).

In our example, the VA1B is the target DB2 system. Issue SSID VA1B from the option line of DB2 Administration Menu (ADB2) (Figure 4-26).

```
ADB2 dmin ----- DB2 Administration Menu 10.1.0 ----- 16:50
Option ==> SSID VA1B

      1 - DB2 system catalog                DB2 System: VA1A
      2 - Execute SQL statements            DB2 SQL ID: SYSADM
      3 - DB2 performance queries          Userid   : SYSADM
      4 - Change current SQL ID             DB2 Schema: SYSADM
      5 - Utility generation using LISTDEFS and TEMPLATES DB2 Rel   : 1015
      P - Change DB2 Admin parameters
      DD - Distributed DB2 systems
      E - Explain
      Z - DB2 system administration
      SM - Space management functions
      W - Manage work statement lists
      X - Exit DB2 Admin
      CC - DB2 catalog copy version maintenance
      CM - Change management

                                                    More:      +

Interface to other DB2 products and offerings:
      I  DB2I

      C  DB2 Object Comparison Tool
```

Figure 4-26 SSID primary command

Note that the subsystem ID on the DB2 Administration Menu panel (ADB2) is now set to VA1B (Figure 4-27).

```
ADB2 dmin ----- DB2 Administration Menu 10.1.0 ----- 16:50
Option ==>

    1 - DB2 system catalog                DB2 System: VA1B
    2 - Execute SQL statements            DB2 SQL ID: SYSADM
    3 - DB2 performance queries          Userid   : SYSADM
    4 - Change current SQL ID             DB2 Schema: SYSADM
    5 - Utility generation using LISTDEfs and TEMPLATES DB2 Rel   : 1015
    P - Change DB2 Admin parameters
    DD - Distributed DB2 systems
    E - Explain
    Z - DB2 system administration
    SM - Space management functions
    W - Manage work statement lists
    X - Exit DB2 Admin
    CC - DB2 catalog copy version maintenance

EeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeN
e Database 2 Administration Tool                                     e   +
e 5655-W34 Copyright IBM Corporation 1995, 2010.                  e
e All rights reserved. Licensed materials - property of IBM.      e
e US Government Users Restricted Rights - Use, duplication or disclosure e
e restricted by GSA ADP schedule contract with IBM Corp.          e
DeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeM
```

Figure 4-27 Switch over to VA1B subsystem

Now we are working on the DB2 catalog on VA1B subsystem and we want to get a list of the stored procedures on the subsystem to verify that the NSP_MEM_ID stored procedure with the V10152010 version ID was created on the subsystem VA1B.

Enter 1 on the DB2 Administration Menu panel (ADB2). The System Catalog panel (ADB21) opens (Figure 4-28).

```
ADB2 dmin ----- DB2 Administration Menu 10.1.0 ----- 16:50
Option ==> 1

      1 - DB2 system catalog                      DB2 System: VA1B
      2 - Execute SQL statements                  DB2 SQL ID: SYSADM
      3 - DB2 performance queries                Userid   : SYSADM
      4 - Change current SQL ID                  DB2 Schema: SYSADM
      5 - Utility generation using LISTDEFS and  DB2 Rel   : 1015
          TEMPLATES
      P - Change DB2 Admin parameters
      DD - Distributed DB2 systems
      E - Explain
      Z - DB2 system administration
      SM - Space management functions
      W - Manage work statement lists
      X - Exit DB2 Admin
      CC - DB2 catalog copy version maintenance
      CM - Change management

                                          More:      +

Interface to other DB2 products and offerings:
  I  DB2I

  C  DB2 Object Comparison Tool
```

Figure 4-28 DB2 Admin Tool main panel on VA1B system

Enter 0 at the option line and specify a name starting with NSP_FIND on the System Catalog panel (ADB21) (Figure 4-29).

```
ADB21 min ----- VA1B System Catalog ----- 16:51
Option ==> 0

Object options:
AO - Authorization options
G - Storage groups
D - Databases
S - Table spaces
T - Tables, views, and aliases
V - Views
A - Aliases
Y - Synonyms
X - Indexes
C - Columns
N - Constraints
DS - Database structures
PDC - DB2 pending definition changes

P - Plans
L - Collections
K - Packages
M - DBRMs
H - Schemas
E - User defined data types
F - Functions
O - Stored procedures
J - Triggers
Q - Sequences
DSP - DS with plans and packages

More: +
DB2 System: VA1B
DB2 SQL ID: SYSADM

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name ==> NSP_FIND > Grantor ==> >
Owner ==> > Grantee ==> >
In D/L/H ==> > Switch Catalog Copy ==> N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
```

Figure 4-29 Search stored procedure name starting with NSP_FIND

Press Enter. The Stored Procedure panel (ADB21O) opens and shows the Stored procedure named NSP_FIND_MEMID (Figure 4-30).

```
ADB21O in ----- VA1B Stored Procedures ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Commands: GRANT
Line commands:
AH - Schema Auth A - Auth DROP - Drop AL - Alter K - Package PA - Params
DIS - Display STO - Stop STA - Start GR - Grant COM - Comment
? - Show all line commands

Sel Schema Name Version A Lang Params Res Q S P C External
* * * * *
----->-----
ADM2 NSP_FIND_MEMID V1015201 Y SQL 3 0 N M N N
***** END OF DB2 DATA *****
```

Figure 4-30 Located stored procedure NDP_FIND_MEMID

Issue an I line command against the stored procedure to discover detailed information about the stored procedure (Figure 4-31).

```
ADB210 in ----- VA1B Stored Procedures ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT
Line commands:
AH - Schema Auth A - Auth DROP - Drop AL - Alter K - Package PA - Parms
DIS - Display STO - Stop STA - Start GR - Grant COM - Comment
? - Show all line commands

                                     S
                                     Res  Q S P C External
Sel  Schema  Name                Version  A Lang Parms  Set  O L R T R Name
   *      *
----->-----
i    ADMR2    NSP_FIND_MEMID      V1015201 Y SQL      3    0 N M N  N
***** END OF DB2 DATA *****
```

Figure 4-31 Issue I line command against stored procedure

Verify the information of stored procedure on the ADB210I1 panel. A native SQL stored procedure, ADMR2.NSP_FIND_MEMID with V10152010, is the active version, the stored procedure is enabled for DDEBUG MODE, and the WLM environment for debugging is WLMENV1 (Figure 4-32).

```

ADB210I1 ----- VA1B Interpretation of an Object in SYSROUTINES ----- 18:11
Option ==>

More:      +

Details for stored procedure: ADMR2.NSP_FIND_MEMID

Schema . . . . . : ADMR2           Name . . . . . : NSP_FIND_MEMID
Owner . . . . . : SYSADM          Specific name . : NSP_FIND_MEMID
Owner type . . . : Auth ID        Created in DB2Ver: 0
Origin . . . . . : N - Native SQL DB information . . : No
Created by . . . : SYSADM         Collection . . . : ADMR2
Number of parms : 3              External name . . :
Language . . . . : SQL           Called on null . . : Y
Parameter style :                SQL data access  : M - Modifies SQL
Stay resident . . : No           CPU service units: 0
WLM environment  : WLMENV1       Program type . . :
WLM for nested . :              LOB columns . . . : 0
External security: D - DB2 AS user Commit on return : No
Result sets . . . : 0           Created TS   : 2010-10-15-16.47.51.669709
Fenced . . . . . :              Altered TS    : 2010-10-15-16.47.51.669709
Parameter CCSID  : 37 - US EBCDIC Failure action : S - Stop
Deterministic . . : N           Consistency token: 18D77EA703CA45D5
Runtime options  :              Version . . . . : V10152010
Active version . . : Yes        Debug mode . . . : Enabled
Package path . . :

Associated remarks :

```

Figure 4-32 Interpretation of stored procedure

Press F3 to go back to the Stored Procedures panel (ADB210).

DB2 implicitly creates a package when a stored procedure is created. We can look up the package of the stored procedure by issuing a K line command next to the stored procedure. In our example, we enter the K line command against the stored procedure NSP_FIND_MEMID (Figure 4-33).

```
ADB210 in ----- VA1B Stored Procedures ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT
Line commands:
AH - Schema Auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment
? - Show all line commands

                                     S
                                     Res  Q S P C External
Sel  Schema  Name                Version  A Lang Parms  Set  O L R T R Name
   *      *
----->-----
K    ADMR2    NSP_FIND_MEMID      V1015201 Y SQL        3   0 N M N  N
*****
                                END OF DB2 DATA *****
```

Figure 4-33 Issue K line command to stored procedure

Press Enter. The packages panel (ADB21K) opens (Figure 4-34). The collection ID of the package is ADMR2, OWNER is SYSADM, QAULIFER is SYSADM, and the Version ID is V10152010, which were the values that we specified in the BIND SQL Procedure Package panel (ADB21KB) when we used a BIND DEPLOY on the NSP_FIND_MEMID stored procedure in Figure 4-23 on page 98.

```
ADB21K in ----- VA1B Packages ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: BIND REBIND FREE VERSIONS GRANT ALL PLANMGMT
Line commands:
DP - Depend  A - Auth  T - Tables  V - Views  X - Indexes  S - Table spaces
Y - Synonyms RB - Rebind F - Free  B - Bind  BC - Bind copy  GR - Grant
EN -Enab/disab con  PL - Package lists  P - Local plans  LP - List PLAN_TABLE
I - Interpret  SQ - SQL in package  VE - Versions  D - Databases  RO - Role

                                     V I V O Quali-  R ED
S  Collection      Name      Owner      Version (trunc)  D S A P fier  L XR
   *              *        *        *              * * * * *  * **
----->-----
ADMR2             NSP_FIND SYSADM  V10152010      R S Y Y SYSADM  C NR
*****
                                END OF DB2 DATA *****
```

Figure 4-34 Package NSP_FIND_MEMID displayed

To see the detailed information of the package, use the I (Interpret) line command. Enter I next to the NSP_FIND_MEMID package (Figure 4-35 on page 107).


```

ADB21K in ----- VA1B Packages ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Commands: BIND REBIND FREE VERSIONS GRANT ALL PLANMGMT
Line commands:
DP - Depend A - Auth T - Tables V - Views X - Indexes S - Table spaces
Y - Synonyms RB - Rebind F - Free B - Bind BC - Bind copy GR - Grant
EN -Enab/disab con PL - Package lists P - Local plans LP - List PLAN_TABLE
I - Interpret SQ - SQL in package VE - Versions D - Databases RO - Role

S Collection          Name      Owner      Version (trunc)  V I V O Quali-  R ED
*                  *          *          *              * * * * *      L XR
----->-----
i ADMR2              NSP_FIND SYSADM   V10152010      R S Y Y SYSADM  C NR
***** END OF DB2 DATA *****

```

Figure 4-35 Issue I line command against the package

Press Enter. The Interpretation of an Object in SYSPACKAGE panel (ADB21KI1) opens and the detailed package information is displayed (Figure 4-36).

```

ADB21KI1 ----- VA1B Interpretation of an Object in SYSPACKAGE -----
Option ==>

Details for package: NSP_FIND > in collection: ADMR2

Package type . . . . . : Native SQL routine package
Version . . . . . : V10152010
Authorization ID of owner . . . : SYSADM
Owner type . . . . . : Auth ID
Authorization ID of creator . . : SYSADM
:
:
Version under which package bound: V10
Qualifier for unqualified SQL . : SYSADM
Operative status of package . . : Package is valid and operative
Resource and authorization check : At RUN time
Size of the base section (bytes) : 4224 (in EDM pool during execution)
Average DML section size (bytes) : 2738 (loaded when needed during exec)
Package bound with EXPLAIN . . . : No
SQLERROR specified at BIND time : No - SQLERROR(NOPACKAGE) specified
BIND or REBIND from remote loc. : Yes - the (RE)BIND was from a remote system
Remote packages creation method :
Source of the package . . . . . : DSN10015.<SYEC3DB2>
Number of enabled/disabled conn. : 0
:
:
:

```

Figure 4-36 Interpretation of NSP_FIND_MEM package

The native stored procedure NSP_FIND_MEMID with version V10152010 has been successfully copied over to the target subsystem.



Universal table space

DB2 9 for z/OS introduced a new type of table space called the *universal table space* (UTS). A universal table space is a combination of the partitioned and segmented table space. You can combine the benefits of segmented table space management with partitioned table space organization by using universal table spaces.

In this chapter, we discuss the two types of universal table space: the partitioned-by-growth table space and the range-partitioned table space (also called partitioned-by-range (PBR)).

We also describe the MAKEPBG and MAKEPBR commands provided by the DB2 Administration Tool. The MAKEPBG command converts the table space to a partitioned-by-growth (PBG) table space. The MAKEPBR command converts the table space to a partitioned-by-range (PBR) table space.

This chapter contains the following topics:

- Universal table space
- DB2 Administration Tool: MAKEPBG

5.1 Universal table space

DB2 table spaces can be exclusively segmented, exclusively partitioned, or both segmented and partitioned. A table space that is exclusively segmented is ideal for storing more than one table. The pages hold segments, and each segment holds records from only one table. A table space that is exclusively partitioned stores a single table. DB2 divides the table space into physical partitions.

DB2 9 for z/OS introduced a new type of table space, the universal table space. A universal table space is a combination of the segmented and partitioned table space schema. Compared to the table spaces that are exclusively partitioned, the universal table space requires more space map pages. However, A segmented space map page provides more information about free space than a regular partitioned space map page. Therefore, it provides better space management relative to varying-length rows. A universal table space also improves mass delete performance because mass delete in a segmented table space tends to be faster than other type of table space. In addition, you can immediately reuse all or most of the segments of a table.

Classic partitioned table spaces are still supported. You can create classic partitioned table spaces by specifying the SEGSIZE clause with a value of 0 on the CREATE TABLESPACE statement.

There are two type of universal table space: the partition-by-growth table space and range-partitioned table space.

5.1.1 Partition-by-growth universal table space

Before DB2 8, partitioned tables required index key ranges to determine the row data placement. In DB2 8, table spaces can be partitioned using table column values. Partitioned tables provide more granular locking and parallel operations by spreading the data over more data sets. Starting with DB2 9 for z/OS, you have the option to partition according to data growth, which enables segmented tables to be partitioned as they grow without the need for specifying key ranges. As a result, segmented tables benefit from increased table space limits and SQL and utility parallelism that were formerly available only to partitioned tables.

With DB2 10 for z/OS, you no longer can create a simple table space. DB2 allows the existing simple table space that was created prior to DB2 10 for z/OS. DB2 creates a partitioned-by-growth table space with NUMPART of 1 when you issue a CREATE TABLESPACE statement with MAXPARTITIONS and without the NUMPARTS clause.

You use the MAXPARTITIONS clause on the CREATE TABLESPACE statement to specify the maximum number of partitions that the partition-by-growth table space can accommodate. You can use the MAXPARTITIONS clause on the ALTER TABLESPACE statement to alter the maximum number of partitions to which an existing partition-by-growth table space can grow. With DB2 10 for z/OS, you also can add partitions by using the ALTER TABLE ADD PARTITION statement for a partition-by-growth universal table space. For more information about DB2 Administration Tool support on ADD PARTITION, see 14.5, “ADD PARTITION” on page 454.

5.1.2 Range-partitioned universal table space

A range-partitioned universal table space uses a segmented table space organization and is based on partitioning ranges. A range-partitioned universal table space contains a single table. You can create an index of any type on a table in a range-partitioned table space.

The range-partitioned table space does not replace the existing classic partitioned table space.

You can create a range-partitioned table space by specifying both the `SEGSIZE` and `NUMPARTS` keywords for the `CREATE TABLESPACE` statement. You can specify partition ranges for a range-partitioned universal table space using a subsequent `CREATE TABLE` or `CREATE INDEX` statement.

With DB2 10 for z/OS, you can create a table space by specifying `NUMPARTS` without specifying the `SEGSIZE` or `MAXPARTITIONS` options. DB2 creates a range-partitioned universal table space with a default table space `SEGSIZE` of 32. A `SEGSIZE` of 32 means 32 pages are to be assigned to each segment of the table space.

DB2 Administration Tool support of universal table space

Some of the DB2 features, such as the `CLONE` table (DB2 9 for z/OS NFM or higher) (see 6.2, “DB2 Administration Tool support for `CLONE` tables” on page 144 and `INLINE LOB` (DB2 10 for z/OS NFM) see 14.2, “Inline LOBs” on page 433), requires that the table resides in a universal table space.

DB2 Administration Tool provides the `MAKEPBG` and `MAKEPBR` commands to allow you to convert a segmented table space or a partitioned table space to a universal table space.

Note: Universal table spaces only allow one table per table space. You encounter `SQLCODE -646` if you try to convert a simple table space that contains multiple tables to a universal table space partitioned-by-growth table space.

5.2 DB2 Administration Tool: MAKEPBG

The `MAKEPBG` command allows you to:

- ▶ Convert the segmented table space to `PBG` table space
- ▶ Convert the classic partitioned table space to `PBG` table space
- ▶ Convert the `PBR` table space to the `PBG` table space

The following example shows the steps of converting a classic partitioned table space to a partitioned-by-growth table space using the DB2 Administration Tool's `MAKEPBG` command

In our example, the table space TPAADMR2 created on a DB2 9 for z/OS subsystem is a classic partitioned table space. To locate the specific table space, enter 1 on the Option line on DB2 Administration Menu panel (ADB2) and press Enter (Figure 5-1).

```
ADB2 dmin ----- DB2 Administration Menu 10.1.0 -----
Option ==> 1

  1 - DB2 system catalog                      DB2 System: V91A
  2 - Execute SQL statements                  DB2 SQL ID: ADMR2
  3 - DB2 performance queries                 Userid   : SYSADM
  4 - Change current SQL ID                   DB2 Schema: ADMR2
  5 - Utility generation using LISTDEFS and  DB2 Rel   : 915
      TEMPLATES
  P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
  E - Explain
  Z - DB2 system administration
  SM - Space management functions
  W - Manage work statement lists
  X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

                                           More:      +

Interface to other DB2 products and offerings:
  I  DB2I

  C  DB2 Object Comparison Tool
```

Figure 5-1 Enter option 1 on DB2 Admin main menu

The DB2 System Catalog panel (ADB21) opens. To locate the TPAADMR2 table space used in our example, enter S at the option line and enter TPAADMR2 into the name field (Figure 5-2). Press Enter.

```

ADB21 min ----- V91A System Catalog -----
Option ==> S

Object options:
A0 - Authorization options
G - Storage groups
D - Databases
S - Table spaces
T - Tables, views, and aliases
V - Views
A - Aliases
Y - Synonyms
X - Indexes
C - Columns
N - Constraints
DS - Database structures

P - Plans
L - Collections
K - Packages
M - DBRMs
H - Schemas
E - User defined data types
F - Functions
O - Stored procedures
J - Triggers
Q - Sequences
DSP - DS with plans and packages

More: +
DB2 System: V91A
DB2 SQL ID: ADMR2

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name ==> TPAADMR2 > Grantor ==> >
Owner ==> > Grantee ==> >
In D/L/H ==> > Switch Catalog Copy ==> N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)

```

Figure 5-2 Locate the TPAADMR2 table space

The table space TPAADMR2 is displayed on the Table Spaces panel (ADB21S). Notice that there is a 2 under the Parts column, which indicates that the table space has two partitions. The Segsize is 0 and there is a blank under the T column, which means that the table space type is not a MEMBER, LARGE, nor LOB table space. The TPAADMR2 table space created on a DB2 9 for z/OS subsystem is a classic partitioned table space (Figure 5-3).

```

ADB21S in ----- V91A Table Spaces ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Commands: GRANT MIG DIS STA STO ALL
Line commands:
T - Tables D - Database A - Auth G - Storage group ICS - Image copystatus
DIS - Display table space STA - Start table space STO - Stop table space
? - Show all line commands

Select Name DB Name Parts Bpool L E S I C Tables Act. pages Segsz T L
* * * * *
-----
TPAADMR2 DBAADMR2 2 BP0 A N A N Y 1 -1 0 Y
***** END OF DB2 DATA *****

```

Figure 5-3 TPAADMR2 table space

You can examine detailed information about the table space by issuing a I (interpret) line command next to the table space. Enter the I line command next to the table space TPAADMR2 and press Enter (Figure 5-4).

```
ADB21S in ----- V91A Table Spaces ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL
Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy
status
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

Select Name      DB Name      Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *          * *      * * * * *      *          * * *
-----
i      TPAADMR2 DBAADMR2      2 BPO  A N A N Y      1      -1      0  Y
*****
***** END OF DB2 DATA *****
```

Figure 5-4 Interpret line command

The Interpretation of an Object in SYSTABLESPACE panel (ADB21SI1) opens (Figure 5-5 on page 115). It contains detailed information about the table space. Some of the key characteristics relevant to our example are:

- Partitions** The number of partitions. Value 0 means it is a simple or segmented table space. > 0 means it is a partitioned table space.
- SEGSIZE** Indicates that the table space is segmented and defines how many pages to assign to each segment.
- Max Partitions** Values 1-4096 specify the maximum number of partitions to which the table space can grow. This type of table space is known as partitioned-by-growth (PBG) table space. Specify 0 for a non-PBG table space.

The key characteristics of the TPAADMR2 table space are:

- ▶ Partitions: Two
- ▶ Segment Size: Not Segm.
- ▶ Table space type: Not Member, Large, or LOB
- ▶ TS allocation status: Table space was explicitly created
- ▶ Max Partitions: Zero

These characteristics show that the TPAADMR2 table space is a user explicitly defined classic partitioned table space.


```

ADB21SI1 ----- V91A Interpretation of an Object in SYSTABLESPACE -----
Option ==>

More:      +

Details for table space : DBAADM2.TPAADM2
TS owner      : ADMR2      TS name      : TPAADM2  Database name : DBAADM2
Created by    : SYSADM     Created Timestamp : 2010-12-01-11.08.47.066093
Max rows/page : 255       Altered Timestamp : 2010-12-01-11.08.47.066093
Descriptor ID : 1         TS ID (PSID)  : 2         Database ID   : 287
Partitions   : 2         Page size KB  : 4         Tables in TS  : 1
Segment size : Not segm. Encod. scheme : E         Lock max      : SYSTEM
SBCS CCSID    : 37        DBCS CCSID   : 0         Mixed CCSID   : 0
Clone         : No        Max partitions: 0

Table space type      : Not MEMBER, Large, or LOB
Table space status      : Available
TS allocation status : Table space was explicitly created
Buffer pool name        : BP0
Lock size for TS        : ANY - Lock scope is determined by DB2
Close rule for data set : Y - Data sets are closed after use
Erase rule for data set : N/A - Table space is partitioned
Log                     : Y - Table space has the LOGGED attribute
Oldest data version     : 0
Current data version    : 0
Data instance number    : 1
Created in DB2 Version  : M - DB2 V9
Creator type            : Auth ID

Statistical data : RUNSTATS timestamp: 0001-01-01-00.00.00.000000
Number of pages used   : -1 (float: -1.0000000000000000E+00)
Allocated space (KB)   : TS is not STOGROUP defined or STOSPACE not run
Average row length     : Statistics have not been gathered

```

Figure 5-5 Interpretation of table space TPAADM2

Press PF3 to go back to the Table Spaces panel (ADB21S).

5.2.1 MAKEPBG

Now we convert the TPAADM2 table space from the classic partitioned table space to an universal partitioned-by-growth table space using the DB2 Administration Tool MAKEPBG command.

To use the MAKEPBG command, we need to first use the ALT command.

ALT is a new function in the DB2 Administration Tool for z/OS V10. It replaces and consolidates the ALC and RDEF functions in the prior release of DB2 Administration Tool. For a detailed description of the ALT command, refer to 7.1, “ALT command” on page 166.

Converting the classic partitioned table space to a PBG table space

On the Table Spaces panel (ADB21S), enter the ALT command next to the TPAADMR2 table space (Figure 5-6) and press Enter.

```
ADB21S in ----- V91A Table Spaces ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL
Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy
status
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

Select Name      DB Name    Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *          * *          * * * * *          *          * * *
-----
ALT    TPAADMR2 DBAADMR2      2 BP0    A N A N Y          1          -1          0    Y
***** END OF DB2 DATA *****
```

Figure 5-6 Issuing the ALT command against TPAADMR2

The Redefine Table Space panel (ADB21SAR) opens. Notice that there are two partitions displayed with the TESTCAT and the VCAT, and SGAADMR2 is the Stogroup name.

There are several primary commands on this panel. We are most interested in the MAKEPBG and MAKEPBR commands, which are to the right of the available primary commands:

- ▶ The MAKEPBG command sets the Numparts field to 0 and the Max Partitions field to > 0 so that you can convert a table space to partition-by-growth (PBG) table space.
- ▶ The MAKEPBR command sets the Numparts field to >0 and SEGSIZE to >0 so that you can convert a table space to partition-by-range (PBR) table space.

To convert the TPAADMR2 table space to partition-by-growth table space, enter MAKEPBG at the Command line or place the cursor under the MAKEPBG command and press Enter (Figure 5-7).

```
ADB21SAR ----- V91A Redefine Table Space ----- Row 1 to 2 of 2
Command ==> MAKEPBG                               Scroll ==> CSR

Commands: CONTINUE ORIGINAL BALANCE VALUES          MAKEPBG MAKEPBR
Line commands: S - Split part  R - Remove part  0 - Original data
                  C - Clear data
CREATE TABLESPACE: TPAADMR2 IN DBAADMR2

Numparts . . . . . 2          Large . . . . . NO      LOB . . . . . NO
Define . . . . . YES          DSSIZE . . . . .        LOG . . . . . YES
Member Cluster . . NO        SEGSIZE . . . . . 0      CCSID . . . . EBCDIC
Buffer Pool . . . . BPO      Close Rule . . YES     Max Rows . . 255
Lock Size . . . . . ANY      Lock Part . . . NO      Lock Max . . SYSTEM
Max Partitions . . 0

S  Part      Pqty      Sqty  Free Pct      E T S
-----
Default:      12        -1    0    5 NO      N Y I TESTCAT  StogroupGBPCache
1
2
***** END OF DB2 DATA *****
```

Figure 5-7 Issuing the MAKEPBG primary command

The Redefine Table Space panel (ADB21SAR) is updated with the new field value for converting to partition-by-growth table space (Figure 5-8).

```
ADB21SAR ----- V91A Redefine Table Space ----- Row 1 to 1 of 1
Command ==>                               Scroll ==> CSR

Commands: CONTINUE ORIGINAL                                MAKEPBG MAKEPBR
Line commands: S - Split part  R - Remove part  0 - Original data
                  C - Clear data
CREATE TABLESPACE: TPAADMR2 IN DBAADMR2  (Convert to Partition-by-Growth)

Numparts . . . . . 0          Large . . . . . NO      LOB . . . . . NO
Define . . . . . YES          DSSIZE . . . . .        LOG . . . . . YES
Member Cluster . . NO        SEGSIZE . . . . . 0      CCSID . . . . EBCDIC
Buffer Pool . . . . BPO      Close Rule . . YES     Max Rows . . 255
Lock Size . . . . . ANY      Lock Part . . . NO      Lock Max . . SYSTEM
Max Partitions . . 4096

S  Part      Pqty      Sqty  Free Pct      E T S
-----
0      24        0    5 NO      N Y I          StogroupGBPCache
***** END OF DB2 DATA *****
```

Figure 5-8 ADB21SAR updated with conversion to PBG information

The (Convert to Partition-by-Growth) message displayed at the right of the Create Table Space: TPAADMR2 in DBAADMR2 line indicates that the command is getting ready to convert the current table space to the PBG table space.

Notice that the Numparts field is changed from 2 to 0, Max Partitions is changed to 4096, which is the default value, SEGSIZ is still 0, the Part field changed from 2 to 0, and the VCAT has been blanked out.

To continue the MAKEPBG process, enter continue at the Command line or place the cursor under the CONTINUE command and press Enter (Figure 5-9).

```

ADB21SAR ----- V91A Redefine Table Space ----- Row 1 to 1 of 1
Command ==> continue                               Scroll ==> CSR

Commands: CONTINUE ORIGINAL                        MAKEPBG MAKEPBR
Line commands: S - Split part  R - Remove part  0 - Original data
               C - Clear data
CREATE TABLESPACE: TPAADMR2 IN DBAADMR2 (Convert to Partition-by-Growth)

Numparts . . . . . 0          Large . . . . . NO    LOB . . . . . NO
Define . . . . . YES         DSSIZE . . . . .      LOG . . . . . YES
Member Cluster . . NO       SEGSIZE . . . . . 0    CCSID . . . . . EBCDIC
Buffer Pool . . . . . BPO    Close Rule . . YES   Max Rows . . 255
Lock Size . . . . . ANY     Lock Part . . . NO    Lock Max . . SYSTEM
Max Partitions . . 4096

               Free Pct      E T S
S  Part      Pqty      Sqty Page Free Compr R M T VCAT      StogroupGBPCache
-  -----
               0          24          0  5 NO  N Y I          SGAADMR2 CHANGED
***** END OF DB2 DATA *****

```

Figure 5-9 Issuing the CONTINUE command to continue the MAKEPBG processing


```

ADBPALT ----- ALTER - Build Analyze and Apply Job-----
Option ==>

Specify the following:

Worklist information:
  Worklist name . . . . . MAKEPBGA (also used as middle qualifier in DSNs)
  Prefix for data sets . . . SYSADM

Data set information:
  PDS final qualifiers . . . 'SYSADM.MAKPBGA.JCL'
  Member name . . . . . MAKPBGA
  Delete member name . . ADBDELET (Optional job to delete work data sets)

Options:
  Generate online . . . . . NO          (Yes/No)
  Generate one job . . . . . YES         (Yes/No)
  Member name or prefix . . APPLY
  As work statement list . . NO          (Yes/No)
  Unload method . . . . . U             (Unload, Parallel unload, HPU)
  Authorization Switch ID . . <NONE>     (SQLID to sign on as, blank or <NONE>)

Optional processes:
  Run CHECK DATA . . . . . YES          (Yes/No)
  Run COPY . . . . . R                 (after: Reload/Alter/Both/None)
  Run REORG/REBUILD . . . . . M         (Mandatory, All relevant, None)
  Run RUNSTATS . . . . . B            (after: Reload/Alter/Both/None)
  Run REBIND . . . . . NO                (Yes/No)

Utility control options:
  Use templates . . . . . NO              (Yes/No)
  Use utility options . . . NO            (Yes/No)

BP - Change batch job parameters
TU - Specify TEMPLATE usage
UO - Customize utility options

```

Figure 5-12 Specify options for building analyze and apply jobs

Press Enter. The Apply Job Data Set panel (ADBPALTJ) opens. You need to specify the data set name of a partitioned data set where the apply job should be stored.

In our example, the 'SYSADM.MAKEPBGA.APPLYJCL' data set name is created by DB2 Administration Tool. Notice that the Worklist name specified on the ALTER - Build Analyze and Apply Job panel (ADBPALT) is used as the middle qualifier of the APPLYJCL (Figure 5-13).

```

ADBPALT ----- ALTER - Build Apply Job -----
Option ==>

Specify the following:

Worklist information:
  Worklist name . . . . . MAKEPBGA (also used as middle qualifier in
DSNs)
EeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeN
e ADBPALTJ ----- Alter - Apply Job Data Set ----- e
e
e Enter/verify the following:
e Data Set Name . . . SYSADM.MAKEPBGA.APPLYJCL
e
e
e
e
DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
  Generate one job . . . . . YES      (Yes/No)
  Member name or prefix . . APPLY
  As work statement list . . NO      (Yes/No)
  Unload method . . . . . U
  Authorization Switch ID . . <NONE>  (SQLID to sign on as, blank or <NONE>)

Optional processes:

```

Figure 5-13 Create the Apply JCL data set

Press Enter. The JOB that performs the analyze portion of the ALTER base change and builds the apply jobs or WSL for the change appears. As you can see in the analyze job, the generated apply job is stored in the data set named 'SYSADM.MAKEPBGA.APPLYJCL' and the TPAADMR2 table space will be dropped and re-created as a partition-by-growth table space with MAXPARTITIONS 4096 defined (Figure 5-14).

```

ISREDDE2  SYSADM.MAKPBGA.JCL(MAKPBGA) - 01.00          Columns 00001 00072
Command ==>                                           Scroll ==> CSR
***** ***** Top of Data *****
==MSG> -CAUTION- Data contains invalid (non-display) characters. Use command
==MSG>           ==> FIND P'.' to position cursor to these
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 //SYSADMD JOB (SETUP),'TESTCASE',
000002 //          RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAM
000003 //          CLASS=A,MSGCLASS=7,REGION=OM,NOTIFY=&SYSUID
000004 //* THIS IS A TEST
000005 //*
000006 //          IF (RC>7) THEN
000007 //          ELSE
000008 //*****ADBSALT**
000009 //*
000010 //* THESE STEPS PERFORM THE ANALYZE PORTION OF THE BASE CHANGE AND
000011 //* BUILDS THE APPLY JOBS OR WSL FOR THE CHANGE.
000012 //*
000013 //* THE APPLY JOBS WILL RESIDE IN:
000014 //*   SYSADM.MAKEPBGA.APPLYJCL
:
:
:
000183 /*
000184 //DDLIN      DD *
000185 ADMIN DROP TABLESPACE DBAADMR2.TPAADMR2;
000186 CREATE TABLESPACE TPAADMR2 IN DBAADMR2 USING STOGROUP SGAADMR2
000187 PRIQTY 24          ERASE NO  FREEPAGE 0  PCTFREE 5  TRACKMOD YES  GBPCACH
000188 E CHANGED  BUFFERPOOL BPO  LOCKSIZE ANY  LOCKMAX SYSTEM  MAXPARTITIONS 4
000189 096  CLOSE YES  COMPRESS NO  CCSID EBCDIC  MAXROWS 255;
000190 /*
000191 //          ENDIF
000192 //          IF (RC>7) THEN
000193 //          ELSE
000194 //*****ADB2SPFB**
:
:
:

```

Figure 5-14 Analyze job of the ALT of TPAADMR2

Submit the Analyze job and go to the SDSF to examine the job output.

At the SDSF STATUS DISPLAY output queue, locate the analyze job and enter ? next to the analyze job output (Figure 5-15). Press Enter.

SDSF STATUS DISPLAY ALL CLASSES									
COMMAND INPUT ==>									
LINE 62-79 (79)									
SCROLL ==> CSR									
NP	JOBNAME	JobID	Owner	Prty	Queue	C	Pos	SAff	ASys Status
?	MAKEPBGA	JOB00148	SYSADM	1	PRINT	A	51		

Figure 5-15 Enter ? next to the ALT analyze job

The panel for the SDSF JOB DATASET for the Analyze job opens. Enter S next to the REPORT under the DDNAME (Figure 5-16). Press Enter.

The report contains the information about the comparison of the before and after state of the object and the change actions that ADM plans to take to make the modification.

SDSF JOB DATA SET DISPLAY - JOB MAKEPBGA (JOB00148)									
COMMAND INPUT ==>									
LINE 1-11 (11)									
SCROLL ==> CSR									
NP	DDNAME	StepName	ProcStep	DSID	Owner	C	Dest	Rec-CntPage	
	JESMSGLG	JES2		2	SYSADM	7	LOCAL	58	
	JESJCL	JES2		3	SYSADM	7	LOCAL	239	
	JESYSMSG	JES2		4	SYSADM	7	LOCAL	359	
	SYSTSPRT	ALTER		107	SYSADM	7	LOCAL	8	
	ADBPRINT	ALTER		109	SYSADM	7	LOCAL	128	
s	REPORT	ALTER		110	SYSADM	7	LOCAL	138	
	ADBDIAG	ALTER		111	SYSADM	7	LOCAL	46	
	SYSOUT	ALTER		113	SYSADM	7	LOCAL	232	
	ISPLOG	ADBGAJOB		114	SYSADM	7	LOCAL	6	
	SYSTSPRT	ADBGAJOB		116	SYSADM	7	LOCAL	64	
	SYSPRINT	ADBGAJOB		117	SYSADM	7	LOCAL	31	

Figure 5-16 Enter S next to the REPORT

As seen in the analysis REPORT, the DBAADM2.TPAADM2 table space is changed from partitioned (<normal>) to partition-by-growth (PBG) by dropping and re-creating the table space. The MAXPARTITIONS of the table space is 4096 and the DSSIZE is 4 GB (Figure 5-17).

```

Compare stogroup source(SGAADM2) and target(SGAADM2)
  No changes to stogroup

Compare tablespace source(DBAADM2.TPAADM2) and target(DBAADM2.TPAADM2)
  (D)Field CREATOR changed from ADM2 to SYSADM
  (D)Tablespace change from partitioned to partition-by-growth
  (A)Field MAXPARTITIONS changed from 0 to 4096
  (D)Field Tablespace TYPE changed from <normal> to PBG
  (A)Field DSSIZE changed from 0G to 4G
Tablespace DBAADM2.TPAADM2 will be dropped
Tablespace will be recreated

Compare table source(ADM2.TBADD304) and target(ADM2.TBADD304)
  Source and/or target tablespaces use table-controlled partitioning
  Comparison results may not reflect actual object differences
  Tables have identical column lists
  (D)Target table is table partitioned, source table is not
  Table ADM2.TBADD304 will be dropped by dropping the tablespace
  Table will be recreated

Compare index source(ADM2.IUADD304) and target(ADM2.IUADD304)
  Index ADM2.IUADD304 will be dropped by dropping the tablespace
  Index will be recreated because the base table will be dropped and recreated

```

Figure 5-17 Compare REPORT of the ALT analyze job

Keep in mind that this is the analyze portion of the change that was executed. The actual change is not implemented until the APPLY job is executed successfully.

```
ADB27CA n ----- VA1A Alter Tables ----- Row 1 to 1 of 1
EeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeN
e ADB2CONF -- VA1A Alter Tables - Exit Confirmation ----- 16:38 e
e
e You have pressed the End key. Continuing the End action will e
e result in losing any changes made to tables and related objects. e
e
e
e
e Select a choice e
e 1 1. Continue End and lose changes e
e 2. Cancel End to preserve changes e
e
e
e
e
e
e
e
e
e
e
DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
-- -----> -----> -----
DBAADMR2 TPAADMR2 S DBAADMR2 TPAADMR2 NA NA MODIFY
***** END OF DB2 DATA *****
```

The Table Spaces panel (ADB21S) opens. Enter refresh at the Command line to refresh the field values displayed on the panel (Figure 5-19).

```

ADB21S in ----- VA1A Table Spaces ----- Row 1 to 1 of 1
Command ==> refresh Scroll ==> CSR

Commands: GRANT MIG DIS STA STO ALL
Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy
status
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

Select Name      DB Name      Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *          * *      * * * * *      *          *          * * *
-----
*LT      TPAADMR2 DBAADMR2      2 BP0      A N A N Y      1          -1          0      Y
***** END OF DB2 DATA *****

```

126 Managing IBM DB2 10 for z/OS Using the IBM DB2 Administration Tool for z/OS Version 10

After the **refresh** command is entered, the value under the column T, which shows the type of the table space, is changed from blank (means normal) to G (partition-by-growth table space) (Figure 5-20). Notice that there is a 1 under Parts, which indicates that a minimum of one part is defined for the table space.

```

ADB21S in ----- VA1A Table Spaces ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL
Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy
status
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

Select Name      DB Name    Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *          *  *      * * * * *      *          *   * * *
-----
      TPAADMR2 DBAADMR2      1 BP0    A N A N Y      1          12    32 G Y
*****
***** END OF DB2 DATA *****

```

Figure 5-20 Refreshed TPAADMR2 shows type G

To see detailed information about the table space, enter the **i** (interpret) line command next to the TPAADMR2 table space and press Enter (Figure 5-21).

```

ADB21S in ----- VA1A Table Spaces ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL
Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy
status
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

Select Name      DB Name    Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *          *  *      * * * * *      *          *   * * *
-----
i      TPAADMR2 DBAADMR2      1 BP0    A N A N Y      1          12    32 G Y
*****
***** END OF DB2 DATA *****

```

Figure 5-21 Enter the interpret line command next to TPAADMR2

The Interpretation of an Object in SYSTABLESPACE panel (ADB21SI1) opens. It contains detailed information about the converted TPAADMR2 table space.

After the MAKPBG command successfully executes, the table space type of the TPAADM2 now is a G - Partitioned by growth universal table space with a Max partitions of 4096 and a Segment size of 32 (Figure 5-22).

```

ADB21SI1 ----- VA1A Interpretation of an Object in SYSTABLESPACE -----
Option ==>

More:      +

Details for table space : DBAADM2.TPAADM2
TS owner      : SYSADM      TS name      : TPAADM2      Database name : DBAADM2
Created by    : SYSADM      Created Timestamp : 2010-12-10-16.36.41.074337
Max rows/page : 255        Altered Timestamp : 2010-12-10-16.36.41.074337
Descriptor ID : 1          TS ID (PSID)  : 2            Database ID   : 297
Partitions   : 1          Page size KB   : 4            Tables in TS  : 1
Segment size : 32        Encod. scheme : E            Lock max      : SYSTEM
SBCS CCSID    : 37         DBCS CCSID   : 0            Mixed CCSID   : 0
Clone         : No         Max partitions: 4096

Table space type      : G - Partitioned by growth
Table space status    : Available
TS allocation status  : Table space was explicitly created
Buffer pool name      : BPO
Lock size for TS      : ANY - Lock scope is determined by DB2
Close rule for data set : Y - Data sets are closed after use
Erase rule for data set : N/A - Table space is partitioned
Log                   : Y - Table space has the LOGGED attribute
Oldest data version   : 0
Current data version  : 0
Created in DB2 Version : 0
Creator type          : Auth ID

Statistical data : RUNSTATS timestamp: 2010-12-10-16.36.51.463069
Number of pages used  : 12 (float: 1.2000000000000000E+01)
Allocated space (KB)  : 48
Average row length    : 0

```

Figure 5-22 Detailed information about the converted TPAADM2 table space

This concludes our example of the using the DB2 Administration Tool AMAKEPBG function.

As we discussed earlier, to use some of the DB2 features such as CLONE table (DB2 9 for z/OS NFM or higher) (see 6.2, “DB2 Administration Tool support for CLONE tables” on page 144 and INLINE LOB (DB2 10 for z/OS NFM) see 14.2, “Inline LOBs” on page 433) requires that the table resides in a universal table space. Thus, in addition to converting the classic partitioned table space to a partition-by-growth table space, you might choose to convert the classic partitioned table space to a range-partitioned universal table space and maintain the partitioning key at the same time.

5.2.2 MAKEPBR

The DB2 Administration Tool MAKEPBR command allows you to convert a table space to a range-by-partition universal table space.

The MAKEPBR command allow you to:

- Convert the classic segmented table space to PBR table space
- Convert the classic partitioned table space to PBR table space
- Convert the PBG table space to the PBR table space

To convert the classic segmented table space or PBG table space to the partition-by-range universal table space, you need to specify the range values for each of the partitions.

The following example shows the steps of converting a classic partitioned table space to a partitioned-by-range table space using the DB2 Administration Tool MAKEPBR command

In our example, we use the same table space, TPAADMR2, which is a classic partitioned table space, in our MAKEPBG example. Instead of converting the classic partitioned table space to a partition-by-growth universal table space, we convert TPAADMR2 to a range-partitioned universal table space while *maintaining the original partitioning keys*.

Before modifying the table space, take a look at the TPAADMR2 table space and the partitioning key of the table in the table space.

On the DB2 Administration Menu panel (ADB2), enter 1 at the Option line and press Enter (Figure 5-23).

```
ADB2 dmin ----- DB2 Administration Menu 10.1.0 -----
Option ==> 1

  1 - DB2 system catalog                      DB2 System: VA1A
  2 - Execute SQL statements                  DB2 SQL ID: ADMR2
  3 - DB2 performance queries                Userid   : SYSADM
  4 - Change current SQL ID                  DB2 Schema: ADMR2
  5 - Utility generation using LISTDEFS and  DB2 Rel   : 1015
    TEMPLATES
  P - Change DB2 Admin parameters
  DD - Distributed DB2 systems
  E - Explain
  Z - DB2 system administration
  SM - Space management functions
  W - Manage work statement lists
  X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

More:      +

Interface to other DB2 products and offerings:
  I  DB2I

  C  DB2 Object Comparison Tool
```

Figure 5-23 Enter option 1 on the ADB2 panel

The DB2 system Catalog panel (ADB21) opens. To locate the TPAADMR2 table space used in our example, enter S at the option line and specify TPAADMR2 at the name field (Figure 5-24). Press Enter.

```

ADB21 min ----- VA1A System Catalog -----
Option ==> s

Object options:
A0 - Authorization options
G - Storage groups
D - Databases
S - Table spaces
T - Tables, views, and aliases
V - Views
A - Aliases
Y - Synonyms
X - Indexes
C - Columns
N - Constraints
DS - Database structures
PDC - DB2 pending definition changes

P - Plans
L - Collections
K - Packages
M - DBRMs
H - Schemas
E - User defined data types
F - Functions
O - Stored procedures
J - Triggers
Q - Sequences
DSP - DS with plans and packages

More: +
DB2 System: VA1A
DB2 SQL ID: ADMR2

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name ==> TPAADMR2 > Grantor ==> >
Owner ==> > Grantee ==> >
In D/L/H ==> > Switch Catalog Copy ==> N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)

```

Figure 5-24 Locate the TPAADMR2 table space

The Table Spaces panel (ADB21S) opens and shows the summarized information about the TPAADMR2 table space. Notice that there is a 2 under the Parts column, which indicates that the table contains two partitions, and a blank under the Table space type column, which indicates that the TPAADMR2 is a normal table space, and not a LOB, XML, PBG, or PBR table space (Figure 5-25).

```

ADB21S in ----- VA1A Table Spaces ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Commands: GRANT MIG DIS STA STO ALL
Line commands:
T - Tables D - Database A - Auth G - Storage group ICS - Image copy
status
DIS - Display table space STA - Start table space STO - Stop table space
? - Show all line commands

Select Name DB Name Parts Bpool L E S I C Tables Act. pages Segsz T L
* * * * *
-----
TPAADMR2 DBAADMR2 2 BP0 A N A N Y 1 -1 0 Y
***** END OF DB2 DATA *****

```

Figure 5-25 Normal table space TPAADMR2

Let us take a look at the partitioning key of the table that resides in the TPAADMR2 table space.

To see the partition limit key of the table, enter `t` next to the TPAADMR2 table space and press Enter (Figure 5-26).

```
ADB21S in ----- VA1A Table Spaces ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Commands: GRANT MIG DIS STA STO ALL
Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy
status
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

Select Name      DB Name      Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *          * *      * * * * *      *          * * *
-----
T      TPAADMR2 DBAADMR2      2 BP0    A N A N Y      1          -1      0  Y
***** END OF DB2 DATA *****
```

Figure 5-26 Enter `T` next to the TPAADMR2 table space

The Tables, Views, and Aliases panel (ADB21T) opens. The ADMR2.TBADD304 table is the table that resides in the TPAADMR2 table space. LKEY is a line command that displays the limit key of the partition.

Enter the LKEY line command next to the TBADD304 table on the ADB21T panel and press Enter (Figure 5-27).

```
ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Commands: GRANT MIG ALL
Line commands:
  C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
  V - Views    T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
  ? - Show all line commands

Sel   Name          Schema      T DB Name  TS Name      Cols      Rows ChksC
      *            *          * *      *          *          *      *   **
-----
LKEY  TBADD304      ADMR2      T DBAADMR2 TPAADMR2      6          -1      0
***** END OF DB2 DATA *****
```

Figure 5-27 Issue LKEY line command against table TBADD304

The Display Limit Key Values panel (ADB21TAV) opens. This panel shows the limit key values of the table-controlled partitioning table. It is a display only panel with no update capability. The information indicates that there are two partitions on the table and the partitioning limit key value for the first partition is '5THVARCHAR20','EEEEEE' and the limit key value for the second partition is '2NDVARCHAR20','BBBBBB' (Figure 5-28).

The COLUMNS command shows the partition key columns, data type, and its order.

```
ADB21TAV  ----- VA1A Display Limit Key Values ----- Row 1 to 2 of 2
Command ==>                                         Scroll ==> CSR

Commands: COLUMNS

TABLE . . . : ADMR2.TBADD304

Sel   Part Limit key value
----->
      1 '5THVARCHAR20','EEEEEE'
      2 '2NDVARCHAR20','BBBBBB'
***** END OF DB2 DATA *****
```

Figure 5-28 Partitioning keys of the TBADD304 table displayed

Enter COLUMNS at the Command line or place the cursor under the COLUMNS command and press Enter.

The Columns panel (ADB21XAC) opens. It shows that the partition key of TBADD304 table is composed of (P1 DESC, P2 DESC). The data type of P1 is VARCHAR(20) and P2 is CHAR(6) (Figure 5-29).

```
ADB21XAC  VA1A Columns      Row 1 from 2 e
Command ==>                      Scroll ==> CSR e
                                         e
Column Name      Col Type Length Seq e
-----
P1               VARCHAR      20 D e
P2               CHAR         6 D e
***** Bottom of data ***** e
                                         e
```

Figure 5-29 Partition key column of the TBADD304 table

Press PF3 twice to go back to the Table Space panel (ADB21S).

Now we start the process of converting TPAADMR2 to a range-by-partition universal table space.

Converting the classic partitioned table space to PBR table space

On Table Spaces panel (ADB21S), enter the ALT command next to TPAADMR2 table space (Figure 5-30). Press Enter.

```
ADB21S in ----- VA1A Table Spaces ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Commands: GRANT MIG DIS STA STO ALL
Line commands:
T - Tables D - Database A - Auth G - Storage group ICS - Image copy
status
DIS - Display table space STA - Start table space STO - Stop table space
? - Show all line commands

Select Name      DB Name      Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *          * *      * * * * *      *          *      * * *
-----
ALT    TPAADMR2 DBAADMR2      2 BP0    A N A N Y      1          -1      0    Y
***** END OF DB2 DATA *****
```

Figure 5-30 Enter the ALT command next to TPAADMR2

The Redefine Table Space panel (ADB21SAR) opens. Notice that the Numparts of the table space is 2, Max Partitions is 0, and the SEGSIZE is 0 before the conversion.

To the right of the Commands line is the MAKEPBR command. The MAKEPBR command allows you to convert the table space to a Partition-By-Range (PBR) table space.

Enter MAKEPBR at the Command line or place the cursor under the MAKEPBR command and press Enter (Figure 5-31).

```
ADB21SAR ----- VA1A Redefine Table Space ----- Row 1 to 2 of 2
Command ==> MAKEPBR Scroll ==> CSR

Commands: CONTINUE ORIGINAL BALANCE VALUES MAKEPBG MAKEPBR
Line commands: S - Split part R - Remove part 0 - Original data
               C - Clear data
CREATE TABLESPACE: TPAADMR2 IN DBAADMR2

Numparts . . . . . 2
Define . . . . . YES DSSIZE . . . . . LOG . . . . . YES
Member Cluster . . NO SEGSIZE . . . . . 0 CCSID . . . . . EBCDIC
Buffer Pool . . . . BP0 Close Rule . . YES Max Rows . . 255
Lock Size . . . . . ANY Lock Part . . . NO Lock Max . . SYSTEM
Max Partitions . . 0

Free Pct      E T S
S Part      Pqty      Sqty Page Free Compr R M T VCAT      StogroupGBPCache
-----
Default:      12      -1    0    5 NO    N Y I TESTCAT      SGAADMR2 CHANGED
1
2
***** END OF DB2 DATA *****
```

Figure 5-31 Enter the MAKEPBR command on ADB21SAR

The Redefine Table Space panel (ADB21SAR) is updated and the (Convert to Partition-by-Range) text is moved to the middle of the panel next to the CREATE TABLESPACE: TPAADMR2 IN DBAADMR2 statement.

Notice that the SEGSIZE is changed from 0 to 4, which means the table space is segmented and there are four pages in each segment. In our example, we keep the same number of partitions and the partitioning key values. Thus, the Numparts value remains as 2 and the Max Partitions value remains as 0 (Figure 5-32).

ADB21SAR
----- VA1A Redefine Table Space -----
Row 1 to 2 of 2
Command ==>
Scroll ==> CSR

Commands: CONTINUE ORIGINAL BALANCE VALUES MAKEPBG MAKEPBR
Line commands: S - Split part R - Remove part 0 - Original data
C - Clear data

CREATE TABLESPACE: TPAADMR2 IN DBAADMR2
(
Convert to Partition-by-Range
)

Numparts 2
Define YES
Member Cluster . . NO
Buffer Pool BPO
Lock Size ANY
Max Partitions . . . 0

DSSIZE
SEGSIZE 4
Close Rule . . YES
Lock Part . . . NO

LOB NO
LOG YES
CCSID EBCDIC
Max Rows . . 255
Lock Max . . SYSTEM

Free Pct
Page Free Compr
R M T VCAT

S Part Pqty Sqty

E T S

StogroupGBPCache

Default:
1
2

12
-1
0
5 NO
N Y I
TESTCAT
SGAADMR2 CHANGED

***** END OF DB2 DATA *****

Figure 5-32 Ready to convert to a partition-by-range UTS

Now we are ready to continue the conversion to the PBR UTS process.

Enter Continue on the Redefine Table Space panel (ADB21SAR) or place the cursor under the CONTINUE command and press Enter (Figure 5-33).

```

ADB21SAR ----- VA1A Redefine Table Space ----- Row 1 to 2 of 2
Command ==> CONTINUE                               Scroll ==> CSR

Commands: CONTINUE ORIGINAL BALANCE VALUES          MAKEPBG MAKEPBR
Line commands: S - Split part  R - Remove part  0 - Original data
               C - Clear data
CREATE TABLESPACE: TPAADMR2 IN DBAADMR2 (Convert to Partition-by-Range)

Numparts . . . . . 2                                LOB . . . . . NO
Define . . . . . YES                                DSSIZE . . . . .
Member Cluster . . NO                               SEGSIZE . . . . 4    LOG . . . . . YES
Buffer Pool . . . . BPO                             Close Rule . . YES  CCSID . . . . EBCDIC
Lock Size . . . . . ANY                             Lock Part . . . NO  Max Rows . . 255
Max Partitions . . 0                                Lock Max . . SYSTEM

S  Part      Pqty      Sqty  Page Free Pct      E T S
----->----->----->----->----->----->----->----->----->----->
Default:      12        -1    0    5 NO      N Y I TESTCAT  StogroupGBPCCache
          1
          2
***** END OF DB2 DATA *****

```

Figure 5-33 Continue the PBR UTS conversion process

The Alter Tables panel (ADB27CA) opens. Notice that the Oper column of the table space has a value of MODIFY (Figure 5-34). This value means that the MODIFY action takes place on the TPAADMR2 table space. The possible values of Oper are Modify, NEW, or NONE.

```

ADB27CA n ----- VA1A Alter Tables ----- Row 1 to 1 of 1
Command ==> ALTER                               Scroll ==> CSR

Commands:
  ALTER - generate jobs  ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . .      >      (Table Schema)
Name . . . .      >      (Table Name. ? to look up)

Object  Object
Sel Qual  Name      T DB Name  TS Name      RI RI  FK  Chg
*      *      * *      *      Rels Add Add Rqd Oper
----->----->----->----->----->----->----->----->----->
      DBAADMR2 TPAADMR2      S DBAADMR2 TPAADMR2      NA NA      MODIFY
***** END OF DB2 DATA *****

```

Figure 5-34 The TPAADMR2 table space will be modified

The ADB27C A panel is the central hub of ALT processing. The ALTER command generates the Analyze and Apply job for the object change. In our case, it is the table space conversion. For more details about the ALT command, refer to 7.1, “ALT command” on page 166.

To continue the ALT processing, enter ALTER at the Command line or place the cursor under the ALTER command and press Enter.

The Build Analyze and Apply Job panel (ADBPALT) opens, where you specify the options for building the work statement lists or batch job that implements the ALT change (Figure 5-35).

For the MAKEPBR command, we use ALTMKPBR as the middle qualifier of the generated job data set and the same options used in the MAKEPBG case.

The option field description is similar to the ones in “Converting the classic partitioned table space to a PBG table space” on page 116.

```
ADBPALT ----- ALTER - Build Analyze and Apply Job-----
Option ==>

Specify the following:
More:      +

Worklist information:
  Worklist name . . . . . ALTMKPBR (also used as middle qualifier in DSNs)
  Prefix for data sets . . . SYSADM

Data set information:
  PDS final qualifiers . . . ALTMKPBR.JCL
  Member name . . . . . MAKEPBRA
  Delete member name . . ADBDELET (Optional job to delete work data sets)

Options:
  Generate online . . . . . NO          (Yes/No)
  Generate one job . . . . . YES        (Yes/No)
  Member name or prefix . . APPLY
  As work statement list . . NO          (Yes/No)
  Unload method . . . . . U            (Unload, Parallel unload, HPU)
  Authorization Switch ID . . <NONE>    (SQLID to sign on as, blank or <NONE>)
  SECADM Authorization ID . .          (An ID to sign on as, blank or <NONE>)

Optional processes:
  Run CHECK DATA . . . . . YES        (Yes/No)
  Run COPY . . . . . R                (after: Reload/Alter/Both/None)
  Run REORG/REBUILD . . . . M          (Mandatory, All relevant, None)
  Run RUNSTATS . . . . . A            (after: Reload/Alter/Both/None)
  Run REBIND . . . . . NO             (Yes/No)

Utility control options:
  Use templates . . . . . NO          (Yes/No)
  Use utility options . . . NO        (Yes/No)

BP - Change batch job parameters
TU - Specify TEMPLATE usage
UO - Customize utility options
```

Figure 5-35 Specifying options for generating jobs for MAKEPBR

Press Enter and the Apply Job Data Set panel (ADBPALTJ) opens. Here you can specify the data set name for the apply job to be generated by the ALT processing (Figure 5-36).

```

ADBPALT ----- ALTER - Build Apply Job -----
Option ==>

Specify the following:

Worklist information:
  Worklist name . . . . . ALTMKPBR (also used as middle qualifier in DSNs)
  EsxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxN
  e ADBPALTJ ----- Alter - Apply Job Data Set ----- 21:15 e
  e
  e Enter/verify the following:
  e Data Set Name . . . SYSADM.ALTMKPBR.APPLYJCL
  e
  e
  e
  DxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxM
  Generate one job . . . . . YES      (Yes/No)
  Member name or prefix . . APPLY
  As work statement list . . NO      (Yes/No)
  Unload method . . . . . U
  Authorization Switch ID . . <NONE>  (SQLID to sign on as, blank or <NONE>)
  SECADM Authorization ID . .      (An ID to sign on as, blank or <NONE>)

Optional processes:

```

Figure 5-36 Specifying the data set name for the apply job

Press Enter, and the JOB that performs the analyze portion of the ALTER base change and builds the apply jobs or WSL for the change appears. As you can see in the analyze job, the generated apply job is stored in the data set named SYSADM.MAKEPBR.APPLYJCL (Figure 5-37).

```

ISREDDE2  SYSADM.ALTMKPBR.JCL(MAKEPBR) - 01.00          Columns 00001 00072
Command ==> SUB                                         Scroll ==> CSR
***** ***** Top of Data *****
==MSG> -CAUTION- Data contains invalid (non-display) characters. Use command
==MSG>          ==> FIND P'.' to position cursor to these
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 //MAKEPBR JOB (SETUP),'TESTCASE',
000002 /* RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
000003 /*          RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP
000004 /*          CLASS=A,MSGCLASS=7,REGION=OM,NOTIFY=&SYSUID
000005 /*
000006 /*          IF (RC>7) THEN
000007 /*          ELSE
000008 //*****ADBSALT****
000009 /*
000010 /* THESE STEPS PERFORM THE ANALYZE PORTION OF THE BASE CHANGE AND
000011 /* BUILDS THE APPLY JOBS OR WSL FOR THE CHANGE.
000012 /*
000013 /* THE APPLY JOBS WILL RESIDE IN:
000014 /*      SYSADM.ALTMKPBR.APPLYJCL
000015 /*

```

Figure 5-37 Generated analyze job of MAKEPBR

Submit the analyze job. Check the SDSF output of the analyze job. The change is not implemented until the generated apply job is executed.

After the analyze job executes successfully, use the ISPF SPLIT screen option or start another ISPF session. Enter 3.4 at the option line on the OS/390 Primary Option Menu panel and press Enter. Enter the data set name SYSADM.ALTMKPBR.APPLYJCL and submit the SYSADM.ALTMKPBR.APPLYJCL(APPLY001) apply job.

In this PBR conversion case, the DB2 Administration Tool drops the TPAADMR2 table space and re-creates it as a partition-by-range universal table space while keeping the original partitioning keys.


```
ADB27CA n ----- VA1A Alter Tables ----- Row 1 to 1 of 1
EeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeN
e ADB2CONF -- VA1A Alter Tables - Exit Confirmation ----- 21:20 e
e
e You have pressed the End key. Continuing the End action will e
e result in losing any changes made to tables and related objects. e
e
e
e
e Select a choice e
e 1 1. Continue End and lose changes e
e 2. Cancel End to preserve changes e
e
e
e
e
e
e
DssssssssssssssssssssssssssssssssssssssssssssssssssssssM
-----> -----> -----
      DBAADMR2 TPAADMR2          S DBAADMR2 TPAADMR2          NA NA          MODIFY
***** END OF DB2 DATA *****
```

You should be back to the Table Spaces panel (ADB21S).

```

ADB21S in ----- VA1A Table Spaces ----- Row 1 to 1 of 1
Command ==> refresh                               Scroll ==> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL
Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copystatus
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

Select Name          DB Name      Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *              *            * *          * * * * *      *              * * *
-----
*LT      TPAADMR2  DBAADMR2      2 BP0      A N A N Y      1      -1      0  Y
***** END OF DB2 DATA *****

```

The Table Spaces panel (ADB21S) is updated with the new values that resulted from the conversion of the table space.

Notice that the Segsz column of the TPAADMR2 table space changed from 0 to 4 and the type of the table space changed from blank to R - Range-partitioned (Figure 5-40).

```

ADB21S in ----- VA1A Table Spaces ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL
Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copystatus
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

Select Name      DB Name    Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *          *  *          * * * * *      *          * * *
-----
      TPAADMR2 DBAADMR2      2 BP0    A N A N Y      1      -1      4 R Y
*****
***** END OF DB2 DATA *****

```

Figure 5-40 TPAADMR2 changed to range-partitioned UTS

To see more detailed information about the TPAADMR2 table space, enter the I (Interpret) line command next to TPAADMR2 and press Enter (Figure 5-41).

```

ADB21S in ----- VA1A Table Spaces ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL
Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy
status
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

Select Name      DB Name    Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *          *  *          * * * * *      *          * * *
-----
i      TPAADMR2 DBAADMR2      2 BP0    A N A N Y      1      -1      4 R Y
*****
***** END OF DB2 DATA *****

```

Figure 5-41 Enter the Interpret line command against TPAADMR2

The Interpretation of an Object in SYSTABLESPACE panel (ADB21SI1) opens and shows detailed information about the TPAADMR2 table space. Notice that Partitions is now 2, Segment size is 4, Max partitions is 0, and the Table space type is R (Figure 5-42).

```
ADB21SI1 ----- VA1A Interpretation of an Object in SYSTABLESPACE -----
Option ==>

More:      +

Details for table space : DBAADM2.TPAADMR2
TS owner      : SYSADM      TS name      : TPAADMR2  Database name : DBAADM2
Created by    : SYSADM      Created Timestamp : 2010-12-06-21.34.30.563077
Max rows/page : 255        Altered Timestamp : 2010-12-06-21.34.30.563077
Descriptor ID : 1          TS ID (PSID)  : 2          Database ID   : 297
Partitions   : 2          Page size KB  : 4          Tables in TS  : 1
Segment size : 4          Encod. scheme : E          Lock max     : SYSTEM
SBCS CCSID    : 37         DBCS CCSID   : 0          Mixed CCSID  : 0
Clone         : No         Max partitions: 0

Table space type      : R - Range-partitioned universal table space
Table space status      : Available
TS allocation status    : Table space was explicitly created
Buffer pool name        : BP0
Lock size for TS        : ANY - Lock scope is determined by DB2
Close rule for data set : Y - Data sets are closed after use
Erase rule for data set : N/A - Table space is partitioned
Log                     : Y - Table space has the LOGGED attribute
Oldest data version     : 0
```

Figure 5-42 Detailed information about the converted TPAADMR2

Now that the TPAADMR2 table space has been converted to partition-by-range universal table space, let us make sure that the original partitioning limit key has remained the same.

To check the partitioning key, press PF3 until you get back to the Table Spaces panel (ADB21S) and enter T next to the TPAADMR2 table space and press Enter (Figure 5-43).

```
ADB21S in ----- VA1A Table Spaces ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL
Line commands:
T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copystatus
DIS - Display table space  STA - Start table space  STO - Stop table space
? - Show all line commands

Select Name      DB Name  Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *       * *      * * * * *      *          *      * * *
-----
T      TPAADMR2 DBAADM2    2 BP0    A N A N Y        1          -1      4 R Y
***** END OF DB2 DATA *****
```

Figure 5-43 Enter T next to TPAADMR2 on ADB21S

The Tables, Views, and Aliases panel (ADB21T) opens and shows the information on table TBADD304, which resides in the TPAADMR2 table space.

Enter the LKEY line command next to the TBADD304 table and press Enter (Figure 5-44).

```
ADB21T in ----- VA1A Tables, Views, and Aliases ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols      Rows ChksC
      *                  *      * *      *      *        *    **
-----
LKEY TBADD304            ADMR2   T DBADMR2 TPAADMR2  6        -1     0
***** END OF DB2 DATA *****
```

Figure 5-44 Enter LKEY line command next to TBADD304

The Display Limit Key Values panel (ADB21TAV) opens. This panel shows the limit key values of a table-controlled partitioning table. The information indicates that there are two partitions on the table, the partitioning limit key value for the first partition is '5THVARCHAR20','EEEEEE', and the limit key value for the second partition is '2NDVARCHAR20','BBBBBB'. The original partitioning keys were re-created successfully (Figure 5-45).

```
ADB21TAV ----- VA1A Display Limit Key Values ----- Row 1 to 2 of 2
Command ==> Scroll ==> CSR

Commands: COLUMNS

TABLE . . . : ADMR2.TBADD304

Sel  Part Limit key value
----->
      1 '5THVARCHAR20','EEEEEE'
      2 '2NDVARCHAR20','BBBBBB'
***** END OF DB2 DATA *****
```

Figure 5-45 Partitioning key values for the TBADD304 table

The TPAADMR2 table space, a classic partitioned table space with two partitions and partitioning keys of '5THVARCHAR20','EEEEEE' for the first partition and '2NDVARCHAR20','BBBBBB' for the second partition was converted to a partition-by-range universal table space using the DB2 Administration Tool MAKEPBR function.



CLONE tables

CLONE table support was delivered in DB2 9 for z/OS. This solution is designed to address high availability issues when loading data.

In this chapter, we describe CLONE table implementation in the DB2 Administration Tool.

The chapter contains the following topics:

- ▶ Basic concepts of CLONE tables
- ▶ DB2 Administration Tool support for CLONE tables
 - Catalog Navigation
 - ADD a CLONE table
 - DROP a CLONE table
 - EXCHANGE DATA
 - DB2 utility support
- ▶ DB2 utilities and CLONE tables
- ▶ Other DB2 commands and CLONE

6.1 Basic concepts of CLONE tables

There are two types of tables involved in a CLONE relationship: BASE and CLONE.

A CLONE table is an exact physical replica of a BASE table. These two table types share the same:

- ▶ Universal table space
- ▶ Indexes

CLONE tables are created (ADDED) using the ALTER ... ADD CLONE syntax.

There are some restrictions for when tables are involved in a CLONE relationship.

- ▶ You cannot RENAME a BASE table.
- ▶ You cannot ALTER the definition (DDL) of either table.
- ▶ You cannot change the value of the table space MAXPARTITIONS parameter if it is stored in a partition-by-growth (PBG) table space.

Eliminating a CLONE table requires using the ALTER ... DROP CLONE SQL syntax.

CLONE tables and BASE tables can “exchange” data. The data values are not moved in the exchange process; instead, it is the instance numbers on the DSN that change. Meanwhile, an entry is inserted into the SYSCOPY table recording that the exchange occurred.

6.2 DB2 Administration Tool support for CLONE tables

We now discuss the functions provided by DB2 Administration Tool for CLONE tables support in regards to the following tasks:

- ▶ Catalog navigation
- ▶ ADDing a CLONE table
- ▶ DROP a CLONE table
- ▶ EXCHANGE data

6.2.1 Catalog navigation

CLONE tables are stored in SYSIBM.SYSTABLES as a type C. To get a list of CLONE tables enter T on the Option line located at the top of the System Catalog panel. At the bottom of the panel, enter TYPE next to the Column name keyword, = next to the Operator keyword, and C next to the value keyword (Figure 6-1).

```
DB2 Admin ----- DB1S System Catalog ----- 17:47
Option ==> T

                                     More:  -

  G - Storage groups                P - Plans
  D - Databases                    L - Collections
  S - Table spaces                  K - Packages
  T - Tables, views, and aliases    M - DBRMs
  V - Views                        H - Schemas
  A - Aliases                      E - User defined data types
  Y - Synonyms                     F - Functions
  X - Indexes                      O - Stored procedures
  C - Columns                      J - Triggers
  N - Constraints                  Q - Sequences
  DS - Database structures          DSP - DS with plans and packages

Enter standard selection criteria (Using a LIKE operator, criteria saved):
Name      ==>                > Grantor ==>                >
Owner     ==>                > Grantee ==>                >
In D/L/H ==>                > Switch Catalog Copy ==> N   (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column    ==> TYPE          > Operator ==> =             Value ==> C
```

Figure 6-1 List of CLONE tables

BASE and CLONE line commands

To find the related BASE table, enter the BASE line command next to the CLONE table name and press Enter (Figure 6-2).

DB2 Admin ----- DB1S Tables, Views, and Aliases -----							Row 1 from 9
Command ==>							Scroll ==> PAGE
Commands: GRANT MIG ALL							
Line commands:							
C - Columns A - Auth L - List X - Indexes S - Table space D - Database							
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping							
? - Show all line commands							
Sel	Name	Schema	T DB Name	TS Name	Cols	Rows	Chks
*	*	*	* *	*	*	*	*
----->	-----	-----	-----	-----	-----	-----	-----
	GARYTABLE_CLONE	DBA032	C DSN04247	GARYTABL	3	-1	0
	GARYTABLE_CLONE	DNET974	C DSN04269	GARYTABL	3	-1	0
	PAULMTABLE_CLONE	DNET974	C DSN04274	PAULMTAB	3	-1	0
BASE	EM07TB02_SALES_CLO	DBA104	C EMEMMDB7	EM07TB02	9	-1	1
	SCOTTTABLE_CLONE	DBA198	C DSN05123	SCOTTTAB	3	-1	0
	SCOTTABLE_CLONE	DBA198	C DSN05124	SCOTTABL	3	-1	0
	BCLONE	DBA198	C SWHDBV9	NEWBASE	3	-1	0
	SS16EMPC	DBA104	C SSEMMDBF	SS16EMP	19	-1	0
	CLONE4BASETABLE	DNET974	C DSN05762	BASETABL	1	-1	0
***** END OF DB2 DATA *****							

Figure 6-2 Find the related BASE table

The BASE table is displayed on the next panel (Figure 6-3).

DB2 Admin ----- DB1S Tables, Views, and Aliases -----							Row 1 from 1
Command ==>							Scroll ==> PAGE
Commands: GRANT MIG ALL							
Line commands:							
C - Columns A - Auth L - List X - Indexes S - Table space D - Database							
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping							
? - Show all line commands							
Sel	Name	Schema	T DB Name	TS Name	Cols	Rows	Chks
*	*	*	* *	*	*	*	*
-----	-----	-----	-----	-----	-----	-----	-----
	EM07TB02_SALES	DBA104	T EMEMMDB7	EM07TB02	9	-1	1
***** END OF DB2 DATA *****							

Figure 6-3 BASE table display

Issue the CLONE line command next to a BASE table name to display the related CLONE table. If a CLONE table does not exist, the message No rows returned appears below the Command line at the top of the panel. BASE and CLONE are the two line commands that can be issued next to a table name to toggle between the two types of tables sharing the one table space.

Filtering

The typical filtering options may be used to list both object types (Figure 6-4).

```
DB2 Admin ----- DB1S Tables, Views, and Aliases ----- Row 1 from 2
Command ==> Scroll ==> PAGE

Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands

Sel      Name                Schema      T DB Name  TS Name     Cols        Rows Chks
  *              *              * *        *
----->-----
      EM07TB02_SALES      DBA104      T EMEMMDB7 EM07TB02      9          -1      1
      EM07TB02_SALES_CLO DBA104      C EMEMMDB7 EM07TB02      9          -1      1
***** END OF DB2 DATA *****
```

Figure 6-4 List of BASE and CLONE tables

Notice the value under the T (Type) column heading. The BASE table is a type T and the CLONE table is a type C. Also make note that the table space name is the same for both tables. In this particular example, it is the table name that makes the objects unique. A different schema could also be used to provide uniqueness.

T line command and ALL T primary command

Issuing the T (Table) line command next to a table space name that contains a CLONE table displays both the CLONE and BASE tables (Figure 6-5 and Figure 6-6 on page 148).

```
DB2 Admin ----- DB1S Table Spaces ----- Row 1 to 1 of 1
Command ==> Scroll ==> PAGE

Commands: GRANT  MIG  DIS  STA  STO  ALL
Line commands:
  T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copystatus
  DIS - Display table space  STA - Start table space  STO - Stop table space
  ? - Show all line commands

Select Name      DB Name      Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
      *          *          * *      * * * * *      *          *          * * *
-----
t      EM07TB02  EMEMMDB7      1 BP1      R N A Y Y      1          -1          4 G Y
***** END OF DB2 DATA *****
```

Figure 6-5 List tables related to the table space

```

DB2 Admin ----- DB1S Tables, Views, and Aliases ----- Row 1 from 2
Command ==> Scroll ==> PAGE

Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands

Sel   Name                Schema   T DB Name  TS Name   Cols      Rows Chks
   *                *      * *      *
----->-----
      EM07TB02_SALES      DBA104   T EMMMDB7 EM07TB02    9        -1    1
      EM07TB02_SALES_CLO DBA104   C EMMMDB7 EM07TB02    9        -1    1
***** END OF DB2 DATA *****

```

Figure 6-6 List of tables: Result of issuing the T line command

The ALL T (list all tables associated with the list of table spaces) primary command issued on the Command line of the Table Spaces panel also displays both the BASE and CLONE tables (Figure 6-7 and Figure 6-8 on page 149).

```

DB2 Admin ----- DB1S Table Spaces ----- Row 1 to 6 of 6
Command ==> all t Scroll ==> PAGE

Commands: GRANT MIG DIS STA STO ALL
Line commands:
T - Tables D - Database A - Auth G - Storage group ICS - Image copystatus
DIS - Display table space STA - Start table space STO - Stop table space
? - Show all line commands

Select Name      DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
   *      *      * *      * * * * *
-----
      EM07TB01 EMMMDB7    1 BP1    R N A Y Y      1        -1      4 G Y
      EM07TB02 EMMMDB7    1 BP1    R N A Y Y      1        -1      4 G Y
      EM07TB03 EMMMDB7    1 BP1    R N A Y Y      1        -1      4 G Y
      EM07TB04 EMMMDB7    1 BP1    R N A Y Y      1        -1      4 G Y
      EM07TB05 EMMMDB7    1 BP1    R N A Y Y      1        -1      4 G Y
      EM07TB06 EMMMDB7    1 BP1    R N A Y Y      1        -1      4 G Y
***** END OF DB2 DATA *****

```

Figure 6-7 ALL T command

DB2 Admin ----- DB1S Tables, Views, and Aliases -----							Row 1 from 7
Command ==>							Scroll ==> PAGE
Commands: GRANT MIG ALL							
Line commands:							
C - Columns A - Auth L - List X - Indexes S - Table space D - Database							
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping							
? - Show all line commands							
Sel	Name	Schema	T	DB Name	TS Name	Cols	Rows Chks
*	*	*	*	*	*	*	* *
----->-----							
	EM07TB01_ORDERS	DBA104	T	EMEMMDB7	EM07TB01	6	-1 0
	EM07TB02_SALES	DBA104	T	EMEMMDB7	EM07TB02	9	-1 1
	EM07TB02_SALES_CLO	DBA104	C	EMEMMDB7	EM07TB02	9	-1 1
	EM07TB03_ADDRESS	DBA104	T	EMEMMDB7	EM07TB03	4	-1 0
	EM07TB04_CUSTOMERS	DBA104	T	EMEMMDB7	EM07TB04	6	-1 0
	EM07TB05_DETAILS	DBA104	T	EMEMMDB7	EM07TB05	8	-1 0
	EM07TB06_ITEMS	DBA104	T	EMEMMDB7	EM07TB06	6	-1 0
***** END OF DB2 DATA *****							

Figure 6-8 Results of ALL T command

DS and DSP commands

The DS and DSP commands display a hierarchical list of related objects. DSP includes the plans and packages. CLONE tables are displayed in the list with a note indicating that the object is a CLONE (Figure 6-9).

DB2 Admin ----- DB0B Database Structures -----							Row 1 to 12 of 12
Command ==>							Scroll ==> PAGE
Line commands: S - Show object DSN - Data sets							
Sel	Type	Object Name	Qualifier	PSID/		OBID	Note
				DBID	ISOBID		
*	*	*	*	*	*	*	* *
----->-----							
D----		TDTEAM76-----		408	0	0	
	S	TD76TS14	TDTEAM76	408	69	68	UTS - PBG
	T	TD76TB14_DEPT	TEAM76	408	0	70	
	UC	DEPTNO		0	0	0	Primary key
	X	TD76XA14	TEAM76	408	72	71	
	X	TD76XB14	TEAM76	408	74	73	
	X	TD76XC14	TEAM76	408	76	75	
	V	TD76DEPMG1	TEAM76	0	0	0	
	V	TD76DEPT	TEAM76	0	0	0	
	V	TD76EMPDP1	TEAM76	0	0	0	
	V	TD76HDEPT	TEAM76	0	0	0	
	T	TD76TB14_DEPT_CLON	TEAM76	408	0	70	Clone
***** END OF DB2 DATA *****							

Figure 6-9 DS command display

6.2.2 ADDing a CLONE table

The creation of a CLONE table is done by running the ALTER command. To add (create) a CLONE table using the DB2 Administration Tool, enter the AL line command next to a table name.

Note: The table *must* reside in a universal table space.

Enter S (Select) next to the ADD CLONE option on the Alter Table panel (Figure 6-10).

DB2 Admin ----- DB0B Alter Table ----- 19:54	
Command ==>	
Table schema . . . : ADMR3 >	
Table name . . . : TD76TB14_DEPT >	
AUDIT	NONE (None, Changes, or All)
DATA CAPTURE	NONE (None/Changes)
VALIDPROC	NULL (NULL/Program name)
RESTRICT ON DROP . . .	NO (Yes/No)
VOLATILE	NO (Yes/No)
APPEND	NO (Yes/No)
ALTER TABLE with any of the above changes OR select one of the options below	
More: +	
ADD column	ADD MATERIALIZED QUERY
ADD PRIMARY KEY	DROP MATERIALIZED QUERY
DROP PRIMARY KEY	REFRESH MATERIALIZED TABLE
ADD FOREIGN KEY	ADD PARTITIONING KEY
DROP FOREIGN KEY	ADD PARTITION
ADD CHECK constraint	s ADD CLONE
DROP CHECK constraint	DROP CLONE
ADD UNIQUE constraint	ADD VERSIONING

Figure 6-10 AL - ADD CLONE

The Alter Table Add Clone panel opens. To prevent duplicates, you must provide either a new name or a new qualifier (SCHEMA) for the CLONE table. In this scenario, we gave the table a new name (Figure 6-11).

```
DB2 Admin ----- DB0B Alter Table Add Clone ----- 20:17
Command ==>

ALTER TABLE

Table schema . . . ADMR3      >          (? to look up)
Table name . . . . TD76TB14_DEPT      >          (? to look up)

ADD CLONE

Clone table schema . . . ADMR3      >          (? to look up)
Clone table name . . . . TD76TB14_DEPT_CLON >          (? to look up)
```

Figure 6-11 ADD CLONE

Press Enter to generate the ALTER ADD CLONE SQL statement (Figure 6-12).

```
DB2 Admin ----- DB0B Statement Execution Prompt ----- 20:20
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'TEAM76.ALTER.WSL'
Work statement list name ==> ALTER      Action ==> A (Append or Replace)
                                           More:      +

Statement that is about to be executed (first 28 lines):
ALTER TABLE "ADMR3"."TD76TB14_DEPT" ADD CLONE "ADMR3"."TD76TB14_DEPT_CLO
N"
```

Figure 6-12 ALTER ... ADD CLONE syntax

Enter a 1 on the Option line to execute the ALTER stmt. After the CLONE table has been ADDED (created), the ALTER stmt executed message is displayed under the Command line (Figure 6-13).

```

DB2 Admin ----- DB0B Alter Table Add Clone ----- 20:22
Command ==>
ALTER stmt executed

ALTER TABLE

Table schema . . . ADMR3      >          (? to look up)
Table name . . . . TD76TB14_DEPT      >          (? to look up)

ADD CLONE

Clone table schema . . . ADMR3      >  (? to look up)
Clone table name . . . .              >  (? to look up)

```

Figure 6-13 ALTER stmt executed message

6.2.3 DROP a CLONE table

DB2 does not allow any physical changes to be made to tables participating in a CLONE relationship. However, the dropping of a CLONE table can be done by using an ALTER SQL statement.

There are two ways to DROP a CLONE table using the DB2 Administration Tool. The first way is to issue an AL line command next to the name of the BASE table. This results in the Alter Table panel opening (Figure 6-14).

```

DB2 Admin ----- DB1S Tables, Views, and Aliases ----- Row 1 from 7
Command ==>                                         Scroll ==> PAGE

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel   Name                Schema  T DB Name  TS Name   Cols      Rows Chks
*      *                  *      * *        *         *        *   *
----->-----
a1    EM07TB01_ORDERS      DBA104  T EMMMDB7  EM07TB01   6         -1    0
      EM07TB02_SALES      DBA104  T EMMMDB7  EM07TB02   9         -1    1
      EM07TB02_SALES_CLO DBA104  C EMMMDB7  EM07TB02   9         -1    1
      EM07TB03_ADDRESS    DBA104  T EMMMDB7  EM07TB03   4         -1    0
      EM07TB04_CUSTOMERS DBA104  T EMMMDB7  EM07TB04   6         -1    0
      EM07TB05_DETAILS    DBA104  T EMMMDB7  EM07TB05   8         -1    0
      EM07TB06_ITEMS      DBA104  T EMMMDB7  EM07TB06   6         -1    0
***** END OF DB2 DATA *****

```

Figure 6-14 AL line command on BASE table

Select the DROP clone option on the Alter Table panel (Figure 6-15).

```

DB2 Admin ----- DBIS Alter Table -----
09:52
Command ==>

Table schema . . . : DBA104 >
Table name . . . : EM07TB02_SALES >

AUDIT . . . . . NONE      (None, Changes, or All)
DATA CAPTURE . . . . . NONE (None/Changes)
VALIDPROC . . . . . NULL   (NULL/Program name)
RESTRICT ON DROP . . . NO   (Yes/No)
VOLATILE . . . . . NO      (Yes/No)
APPEND . . . . . NO        (Yes/No)

ALTER TABLE with any of the above changes OR select one of the options below

                                     More:  +
ADD column                         ADD MATERIALIZED QUERY
ADD PRIMARY KEY                   DROP MATERIALIZED QUERY
DROP PRIMARY KEY                  REFRESH MATERIALIZED TABLE
ADD FOREIGN KEY                   ADD PARTITIONING KEY
DROP FOREIGN KEY                  ADD/ALTER PARTITION
ADD CHECK constraint              ADD CLONE
DROP CHECK constraint             S DROP CLONE
ADD UNIQUE constraint

```

Figure 6-15 Select DROP CLONE

The Drop Clone panel opens with the DROP CLONE SQL statement at the bottom of the panel (Figure 6-16). Notice that the name of the table is that of the BASE table.

```

DB2 Admin ----- DB0B Drop Clone ----- 20:38
Command ==>

ALTER TABLE

Schema . . . : DBA104 >
Name . . . : EM07TB02_SALES > (? to look up)

DROP CLONE

```

Figure 6-16 DROP CLONE

Another way to DROP a CLONE table is to enter the DROP line command next to the name of the CLONE table (Figure 6-17).

```
DB2 Admin ----- DB1S Tables, Views, and Aliases ----- Row 1 from 3
Command ==> Scroll ==> PAGE

Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols  Rows Chks
    *                   *      * *      *      *      *   *   *
----->-----
      EM07TB01_ORDERS    DBA104  T EMMMDB7  EM07TB01    6      -1    0
      EM07TB02_SALES    DBA104  T EMMMDB7  EM07TB02    9      -1    1
drop EM07TB02_SALES_CLO DBA104 C EMMMDB7 EM07TB02    9      -1    1
***** END OF DB2 DATA *****
```

Figure 6-17 DROP CLONE Table

The Drop Clone panel opens and shows the ALTER TABLE and DROP CLONE syntax (Figure 6-18).

```
DB2 Admin ----- DB0B Drop Clone ----- 20:38
Command ==>

ALTER TABLE

Schema . . . DBA104>
Name . . . . EM07TB02_SALES > (? to look up)

DROP CLONE
```

Figure 6-18 DROP CLONE

Press Enter and the full ALTER ... DROP CLONE SQL syntax is generated and displayed on the Statement Execution Prompt panel (Figure 6-19).

```

DB2 Admin ----- DB1S Statement Execution Prompt ----- 10:02
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
    1 - Execute the statement
    2 - Edit the statement
    3 - Create a batch job with the statement
    4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> DBA104.WORK.STATMNT.LIST
Work statement list name ==> GENCAL      Action ==> A (Append or Replace)
                                                    More:      +

Statement that is about to be executed (first 28 lines):
ALTER TABLE "DBA104"."EM07TB02_SALES" DROP CLONE

```

Figure 6-19 ALTER TABLE DROP CLONE syntax

Press 1 to execute the statement and the CLONE table is dropped.

6.2.4 EXCHANGE data

XCHG is a DB2 Administration Tool line command used to exchange data between a BASE table and a CLONE table. This command can be issued against either table name (Figure 6-20).

```

DB2 Admin ----- DBOB Tables, Views, and Aliases ----- Row 1 from 2
Command ==>                                           Scroll ==> PAGE

Commands: GRANT MIG ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols      Rows ChksC
    *                  *      *  *      *      *        *    **
-----
XCHG TD76TB14_DEPT        TEAM76  T TDTEAM76 TD76TS14    5        -1    0
      TD76TB14_DEPT_CLON TEAM76  C TDTEAM76 TD76TS14    5        -1    0
***** END OF DB2 DATA *****

```

Figure 6-20 Exchange data: XCHG line command

The Exchange Data Between Base and Clone Tables panel opens (Figure 6-21).

```
DB2 Admin -- DBOB Exchange Data Between Base and Clone Tables ----- 17:06
```

```
This command will exchange all data between  
a base table and it's clone table.
```

```
Select a choice
```

1. Execute command
2. Cancel

Figure 6-21 Exchange Data Between Base and Clone Tables

There are two choices: Execute command or Cancel. Enter a 1 to execute or make the exchange. An EXCHANGE DATA SQL statement is generated and displayed on the Statement Execution Prompt panel (Figure 6-22).

```
DB2 Admin ----- DBOB Statement Execution Prompt ----- 17:08  
Option ==>
```

```
DB2 Admin is about to execute the statement below. You have asked to be  
prompted before DB2 Admin executes this type of statement. What do you want to  
do now:
```

- 1 - Execute the statement
- 2 - Edit the statement
- 3 - Create a batch job with the statement
- 4 - Add the statement to the work statement list

```
CAN - Cancel
```

```
Work statement list dsn ==> 'TEAM76.ALTER.WSL'
```

```
Work statement list name ==> ALTER      Action ==> A (Append or Replace)
```

```
More:      +
```

```
Statement that is about to be executed (first 28 lines):
```

```
EXCHANGE DATA BETWEEN TABLE TEAM76.TD76TB14_DEPT AND TEAM76.TD76TB14_DEP  
T_CLON
```

Figure 6-22 EXCHANGE DATA SQL syntax

After the SQL has been executed, the EXCHANGE stmt executed message is displayed under the Command line on the Tables, Views, and Alias panel (Figure 6-23).

```
DB2 Admin ----- DBOB Tables, Views, and Aliases ----- Row 1 from 2
Command ==> Scroll ==> PAGE
EXCHANGE stmt executed
Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands
```

Sel	Name	Schema	T DB Name	TS Name	Cols	Rows	ChksC
*	*	*	*	*	*	*	**
*CHG	TD76TB14_DEPT	TEAM76	T TDTEAM76	TD76TS14	5	-1	0
	TD76TB14_DEPT_CLON	TEAM76	C TDTEAM76	TD76TS14	5	-1	0

```
*****
***** END OF DB2 DATA *****
```

Figure 6-23 EXCHANGE completed

At this point there are two data sets. To see a list of the data set names, navigate to the table space and issue the DSN line command (Figure 6-24).

```
DB2 Admin -- DBOB Data Sets for TS TDTEAM76.TD76TS1 > ----- Row 1 to 2 of 2
Command ==> Scroll ==> PAGE

Line commands: LISTC - List catalog LISTD - List catalog data

Select Data Set Name
-----
DB0BD.DSNDBC.TDTEAM76.TD76TS14.I0001.A001
DB0BD.DSNDBC.TDTEAM76.TD76TS14.I0001.A002
*****
***** END OF DB2 DATA *****
```

Figure 6-24 Data set names for the BASE and CLONE tables

As a result of the “exchange”, the data is now in the data set ending in A002. To confirm this exchange, issue the B or BR (browse) line command next to the name of the BASE table. A No rows returned message should be displayed under the Command line at the top of the panel. Issue the B or BR (browse) line command next to the name of the CLONE table. A list of data values are displayed.

Another source for confirmation is to view the information stored in SYSIBM.SYSCOPY. This can easily be accomplished in the DB2 Administration Tool by issuing the ICS line command next to the table space name (Figure 6-25).

```

DB2 Admin ----- DB0B Databases ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
DIS - Display database  STA - Start database  STO - Stop database  A - Auth
? - Show all line commands

Select Name      Owner      Storage  Buffer      Created      Index
      *          *          Group    Pool        DBID By      T E BPool    I
-----
iCS  TDTEAM76 ADMR1    DSN8G100 BPO          408 ADMR1    E BPO      N
*****
                                END OF DB2 DATA *****

```

Figure 6-25 ICS IC Status

Enter the I (Interpret) line command next to one of the entries that has an “A” under the I (Operation Type) column heading (Figure 6-26).

```

DB2 Admin ----- DB0B Information from SYSIBM.SYSCOPY ---- Row 1 to 14 of 16
Command ==>                                     Scroll ==> PAGE

Line commands:
S - Table space  BR - Browse image copy  L - LISTCAT  LD - LISTCAT data
I - Interpretation  RT - Recover TOCOPY  RO - Recover TOCOPY with options

S  DB Name  PS Name  NUM  I ICDate  ICTime  Devtype  DS Name T
  *      *      *      *  * *      *      *      * *
-----
i  TDTEAM76 TD76TS14      A 101117 171102      TDTEAM76.TD76TS14 T
TDTEAM76 TD76TS14      A 101117 171102      TDTEAM76.TD76TS14 T
TDTEAM76 TD76TS14      C 101011 150049      TDTEAM76.TD76TS14 T
TDTEAM76 TD76TS13      C 101011 150049      TDTEAM76.TD76TS13 T
TDTEAM76 TD76TS12      C 101011 150049      TDTEAM76.TD76TS12 T
TDTEAM76 TD76TS11      C 101011 150049      TDTEAM76.TD76TS11 T
TDTEAM76 TD76TS10      C 101011 150049      TDTEAM76.TD76TS10 T
TDTEAM76 TD76TS09      C 101011 150049      TDTEAM76.TD76TS09 T
TDTEAM76 TD76TS08      C 101011 150048      TDTEAM76.TD76TS08 T
TDTEAM76 TD76TS07      C 101011 150048      TDTEAM76.TD76TS07 T
TDTEAM76 TD76TS06      C 101011 150048      TDTEAM76.TD76TS06 T
TDTEAM76 TD76TS05      C 101011 150048      TDTEAM76.TD76TS05 T
TDTEAM76 TD76TS04      C 101011 150047      TDTEAM76.TD76TS04 T
TDTEAM76 TD76TS03      C 101011 150047      TDTEAM76.TD76TS03 T

```

Figure 6-26 Interpret line command

This command displays detailed information regarding the entry in the SYSCOPY table (Figure 6-27).

```
DB2 Admin ----- DB0B Interpretation of an Object in SYSCOPY ----- 19:30
Option ==>

Recovery details for table space: TD76TS14 In database : TDTEAM76

Data set number within TS . . . : 0
Recovery status change reason . : ALTER
Status change date (yymmdd) . . : 101117 Time (hhmmss) : 171102
Status change timestamp . . . . : 2010-11-17-17.11.02.965894
Log RBA at latest status change : 00007969A38A (Hex)

Data Sharing Member Name . . . : Not in share mode at time of operation

Special recovery considerations : EXCHANGED clone and base

Object of recovery info . . . . : T - Table Space
Lowest partition in range . . . : 0
Highest partition in range . . . : 0
Pages written to copy data set : -1.000000000000000E+00
Pages in TS or I at copy time . : -1.000000000000000E+00
```

Figure 6-27 SYSIBM.SYSCOPY details

At this point, data could be LOADED into the BASE table while applications continue to access the data in the CLONE table.

6.3 DB2 utilities and CLONE tables

The DB2 Administration Tool UTIL line command is not valid when issued next to the name of a CLONE table. It is valid when issued next to the name of the BASE table. A LOAD or UNLOAD utility JCL may be generated.

UTIL is a valid command when issued next to the name of the universal table space shared by the CLONE and BASE tables.

Many of the utilities have a CLONE keyword that, when specified, tells the utility to run on the CLONE table rather than the BASE table. The utilities include:

- ▶ COPY, COPYTOCOPY, MERGECOPY, and MODIFY
- ▶ CHECK INDEX and CHECK DATA
- ▶ REPAIR AUXWARN, REPAIR AUXCHECKPEND, and REPAIR NOREORGPEND
- ▶ REORG, REORG UNLOAD ONLY, ONLINE REORG, and REORG INLINE COPY
- ▶ QUIESCE
- ▶ RECOVER, RECOVER TO RBA, RECOVER LOGONLY, and RECOVER TO LOGPOINT

- REBUILD INDEX
- UNLOAD

To use the CLONE keyword, you must use the Review/change options feature of DB2 Administration Tool. This option is found at the bottom of the panel displayed by the UTIL command (Figure 6-28).

```

DB2 Admin ----- DB0B Table Space Utilities ----- 19:16
Option ==>

Execute utility on                                DB2 System: DB0B
table space TDTEAM76.TD76TS14                    DB2 SQL ID: ADMR1
                                                More: - +

P - Report recovery      Q - Quiesce
R - Runstats             RT - Runstats table all  RR - Runstats report
RX - Runstats (to invalidate dynamic cache)
V - Recover              VC - Recover tocopy      VG - Recover to lastGDG
VI - Rebuild index       VR - Recover torba       VL - Recover logonly
DG - Define GDG for copy data sets                VP - Recover tologpoint
U - Unload

BP - Change batch job parameters

Utility control options:
Review/change options      : YES  (Yes/No)
Generate work statement list : NO   (Yes/No)
Generate template statements : NO   (Yes/No)
Generate modify after copy   : NO   (Yes/No)

```

Figure 6-28 Review /change options

Set Review /change options to YES and a list of parameters for the selected utility are displayed. See Figure 6-29 for an example of the CLONE option when performing an image copy.

```
DB2 Admin ----- DB0B Specify Utility Options - COPY DATA FULL ----- 18:55
Option ==>

Execute utility on table space TDTEAM76.TD76TS14
using the following options:

FULL . . . . . Y          (Yes/No)
CHANGELIMIT . . .          (Yes/No/Any)
  pct-value1 . . .
  pct-value2 . . .
  REPORTONLY . . .          (Yes/No)
PARALLEL . . . . .
  TAPEUNITS . . .
CHECKPAGE . . . . .          (Yes/No)
SYSTEMPAGES . . .          (Yes/No)
CONCURRENT . . . . .          (Yes/No)
SHRLEVEL . . . . .          (R - Reference, C - Change)
CLONE . . . . .          (Yes/No)
SCOPE . . . . .          (A - ALL, P - PENDING)
```

Figure 6-29 COPY CLONE

The CLONE parameter can be set to Yes or No.

6.4 Other DB2 commands and CLONE

The DIS DB2 command shows a status of the BASE and CLONE table. For example, in Figure 6-30, the BASE table shows a type of TSB2 and the CLONE is TSC1.

```
DB2 Admin ----- DB0B Browse DB2 Command Output -- Line 00000000 Col 001 080
Command ==>                                         Scroll ==> PAGE

-DIS DB(TDTEAM76) SPACENAM(TD76TS14) LIMIT(*)

***** Top of Data *****
DSNT360I  -DB0B *****
DSNT361I  -DB0B *   DISPLAY DATABASE SUMMARY
              *   GLOBAL
DSNT360I  -DB0B *****
DSNT362I  -DB0B      DATABASE = TDTEAM76  STATUS = RW
              DBD LENGTH = 28256
DSNT397I  -DB0B
NAME      TYPE PART  STATUS          PHYERRLO PHYERRHI CATALOG  PIECE
-----
TD76TS14  TSB2  0001 RW
          -THRU  0002
TD76TS14  TSB2          RW
TD76TS14  TSC1  0001 RW
          -THRU  0002
TD76TS14  TSC1          RW
***** DISPLAY OF DATABASE TDTEAM76 ENDED *****
DSN9022I  -DB0B DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
```

Figure 6-30 DIS DB



Part 4

Super processes

In this part, we provide more information about the enhancements of the three major functions related to DB2 Administration Tool.

This part contains the following chapters:

- ▶ Chapter 7, “The ALT line command” on page 165
- ▶ Chapter 8, “The MIG line command” on page 213
- ▶ Chapter 9, “DB2 Object Comparison Tool enhancements” on page 235



The ALT line command

The ALT line command provides options when altering the attributes of an object, such as a table, or redefining an object. This command invokes the DB2 Admin Alter ALT function.

After you have defined the required changes, you can use the CONTINUE primary command to generate the batch jobs that perform the actual alter operation.

This chapter contains the following topics:

- ▶ ALT command
- ▶ Using ALT

7.1 ALT command

ALT is a new DB2 Administration Tool for z/OS V10 line command that used to make structural changes to DB2 objects. At the time of the writing of this book, ALT can be used to replace the previous commands shown in Table 7-1.

Table 7-1 ALT command replacements

DB2 Administration Tool Version 7.2 and prior line commands	DB2 Administration Tool for z/OS Version 10 line command	Objects
AL DB RENAME	ALT	Database
AL TS RDEF	ALT	Table space
AL IX RDEF	ALT	Index
ALC TABLE, VIEW, or FOREIGN KEY	ALT	Table, view, or foreign key

7.1.1 Making changes in DB2

Structural changes in DB2 can be categorized as being non-intrusive or intrusive. A non-intrusive change is one where the structure of a DB2 object can be changed using an ALTER statement. An intrusive change, conversely, requires a series of steps that result in the destruction and recreation of the objects.

In earlier releases of the DB2 Administration Tool there were two alter line commands, AL and ALC. The AL line command was valid for all DB2 objects where a structure change could be defined. Only those fields that could be modified by an ALTER statements were presented on the ISPF panel. To make an intrusive change, an additional command had to be issued that would expand the list of parameters. For example, when using the AL command against a table space, the user had the option to redefine (RDEF) the table space. A redefine included the ability to convert a non-partitioned table space to partitioned, or the modification of parameters such as Numparts, Maxpartitions, DSSIZE, and MEMBER CLUSTER, all of which caused DB2 to unload the data, DROP and CREATE the objects, and then load the data. Using the AL command next to a database name presented the user with the option to ALTER some parameters or RENAME the database. The process of renaming a database in DB2 is intrusive and DB2 Administration Tool would generate all of the JCL required to make the change.

The altering of tables worked differently. When the AL command was issued next to a table name, the ISPF panel only displayed those options that could be modified by an ALTER. To make a change that required intrusive actions, the user had to issue the ALC command next to the table name. ALC was a valid command for tables, views, and foreign keys.

In DB2 Administration Tool for z/OS V10, all of the intrusive types of changes are triggered by the ALT command. For example, to RENAME a database, you must issue the ALT line command next to the name of the database. To RDEF a table space, you must issue the ALT line command next to the name of a table space.

ALT, like ALC, supports a smart alter capability, meaning that it generates ALTER statements where possible.

The AL command remains a viable command in DB2 Administration Tool for z/OS V10, but it is restricted to supporting changes that can be accomplished with an ALTER.

```

DB2 Admin ----- DBOB Tables, Views, and Aliases ----- Row 1 from 15
Command ==> Scroll ==> PAGE

Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols      Rows Chks
C
*
-----> -----
-
      TDSPTXT              DSN81010 T DSN8D10P DSN8S10C      3        70      0
      TD76TB01_DEPT        TEAM76   T TDTEAM76 TD76TS01      5        -1      0
*LC  TD76TB02_EMP           TEAM76   T TDTEAM76 TD76TS02     14        -1      2
      TD76TB03_ACT         TEAM76   T TDTEAM76 TD76TS03      3        -1      0
      TD76TB04_PROJ        TEAM76   T TDTEAM76 TD76TS04      8        -1      0
      TD76TB05_PROJACT     TEAM76   T TDTEAM76 TD76TS05      5        -1      0
      TD76TB06_EMPPROJAC   TEAM76   T TDTEAM76 TD76TS06      6        -1      0
      TD76TB07_EACT        TEAM76   T TDTEAM76 TD76TS07      5        -1      0
Esaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
e The ALC line command is no longer supported. Use the ALT command instead. e
DsaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaM
      TD76TB11_EPROJACT    TEAM76   T TDTEAM76 TD76TS11      7        -1      0

```

ANALYZE

ANALYZE determines what actions need to occur to make a change based on the modifications specified by the user. ANALYZE can be managed through the specifications of a set of parameters found on the ALTER Analysis Options panel (Figure 7-2). One of the options on the panel is to “Show this panel prior to each use”. If this is set to YES, then the panel is displayed every time a change is entered. If it is set to NO, then the user has to access the panel in one of the following ways to update the parameters:

- Directly off of the DB2 Administration Tool Menu by entering =P.A on the Command line. If the user elects to set the “Show this panel prior to each use option” to YES, then the panel is displayed every time a change is made.
- Using the OPTIONS keyword on the Alter Tables panel. This command displays the ALTER Analysis Options panel, allowing the user to make changes.

```

DB2 Admin ----- DB0B ALTER Analysis Options ----- 17:25
Option ==>

Please specify the following for DB2 Admin ALTER:

Analysis options:
Run SQLID . . . . . TEAM76 (Blank, an SQLID, or <NONE>)
Use DEFER YES . . . . . NO (Yes/No)
VIEW Column List . . . . . YES (Yes/No)
Perform recovery analysis . . . . . YES (Yes/No)

Perform analysis in batch . . . . . YES (Yes/No)

Show this panel prior to each use . . . YES (Yes/No)

Change diagnostic options . . . . . NO (Yes/No)

```

Figure 7-2 ALTER Analysis Options

A description of the parameters of the ALTER Analysis Options panel is in Table 7-2.

Table 7-2 ALTER Analysis Options panel parameters

Description	Option	Valid values	Comments
Analysis Options	Run SQLID	Blank / SQLID / NONE	Value that is inserted in the CURRENT SQLID statement in the generated APPLY jobs.
	Use DEFER	YES or NO	Used by the CREATE INDEX syntax to delay the building of a new index.
	VIEW Column List	YES or NO	When set to NO, the SELECT * syntax for views is retained.
	Perform Recovery Analysis	YES or NO	When set to YES, a recover set of JCL is generated and provides the ability to back out of completed changes.
Perform Analysis in Batch		YES or NO	ANALYZE may be performed in the foreground or as part of the ADBALTER batch job.

Description	Option	Valid values	Comments
Show this panel prior to each use		YES or NO	For new users, the recommendation is to set this to Yes to display the ALTER Analysis panel for each change. If set to NO, the values used are those which were previously set on the ALTER Analysis Options panel.
Change diagnostic options		YES or NO	Use only if directed by to do so by IBM Technical Support. If set to YES, work files may be defined as permanent.

REPORT

ANALYZE can be run in batch or online. This is controlled by the “Perform analysis in batch” field on the ALTER Analysis panel. If run in batch, reports are generated based on the results of an internal compare process between the object definition stored in the DB2 Catalog and the modifications specified by the user on the ISPF panel. The messaging in the report provides the user with a clear understanding of the differences between the two environments and what actions are going to be taken to make them look the same. The report is routed to SDSF and can be found in the REPORT section of the job output. See Figure 7-3 and Figure 7-4 on page 170 for examples.

DB2 Admin ----- DSNT System Catalog -----
12:26
.
.
Display Filter View Print Options Search Help

SDSF OUTPUT DISPLAY DBA104D JOB05867 DSID 110 LINE 114 COLUMNS 01- 80
COMMAND INPUT ==> SCROLL ==> CSR
Compare stogroup source(DSN8G100) and target(DSN8G100)
No changes to stogroup
Compare database source(TDTEAM76) and target(TDTEAM76)
(A)Field BUFFERPOOL changed from BPO to BP1
Database will be altered
Compare tablespace source(TDTEAM76.TD76TS01) and target(TDTEAM76.TD76TS01)
No changes to Tablespace

Figure 7-3 Object Altering Report

```

DB2 Admin ----- DSNT System Catalog -----
12:26
. . . . .
  Display Filter View Print Options Search Help

-----
SDSF OUTPUT DISPLAY DBA104D JOB05867 DSID 110 LINE 163 COLUMNS 01- 80
COMMAND INPUT ==> SCROLL ==> CSR
Compare table source(Team76.TD76TB01_DEPT) and target(Team76.TD76TB01_DEPT)
  (A)Column LOC_EXAMPLE appended
  Table will be altered

Compare table source(Team76.TD76TB02_EMP) and target(Team76.TD76TB02_EMP)
  Source and/or target tablespaces use table-controlled partitioning
  Comparison results may not reflect actual object differences
  (D)Column CELLPHONE added
  Table Team76.TD76TB02_EMP is partitioned and will be dropped by dropping
tabl
  Table will be recreated
  Table data will not be converted

Compare table source(Team76.TD76TB03_ACT) and target(Team76.TD76TB03_ACT)
  No changes to Table

Compare table source(Team76.TD76TB04_PROJ) and target(Team76.TD76TB04_PROJ)
  No changes to Table

```

Figure 7-4 Object Altering Report (continued)

If the “Perform analysis in batch” option is set to NO, then ANALYZE runs in the foreground. If it is determined that the change can be made using an ALTER, the Alter SQL or Batch panel opens (Figure 7-5). No reports are generated.

```

DB2 Admin -- DSNT Alter - SQL or Batch ----- 19:

The requested changes can all be performed using SQL statements.

Select a choice:
  1 - Perform the SQL statements in foreground
  2 - Generate a batch job

```

Figure 7-5 Execute ALTER using SQL or through batch

APPLY

The term APPLY refers to the process of physically changing the DB2 object. The ADBALTER JCL job generated by ALT builds the APPLY in either a work statement list member or as an MVS JCL job stream in a PDS. Changes are not “applied” to the DB2 Catalog until one of the following events occur:

- ▶ The work statement list member is “run”.
- ▶ The APPLY MVS JCL is submitted.

Basic steps for making a change using ALT

The basic steps for making changes using the DB2 Administration Tool ALT command are as follows.

1. Issue the ALT line command next to the name of a DB2 object. Currently, ALT is a valid line command for database, table space, table, view, and index object types. Refer to number 1 in Figure 7-6.
2. Edit or modify the object definition by entering the changes on the ISPF panel. When all the changes have been entered for an object, the user needs to take action to append the current change to the list of changes located on the bottom of the Alter Tables panel. The action taken depends on the type of object. See number 2 and ACTIONS in Figure 7-6.
 - a. If the object is a database, pressing Enter after entering the changes inserts the change on the Alter Tables panel.
 - b. For a table space or table, the user must enter Continue on the Command line or click the key word **CONTINUE**. The change is inserted or appended to the list of changes on the Alter Tables panel.
 - c. Changing an index works a little differently. There are two panels on which Continue must be entered. The syntax of the index is displayed. The user must press F3 and then the change is inserted into the Alter Tables panel.
 - d. When ALT is used against a view, the DDL of the view is displayed and can be edited. After the change has been specified, pressing F3 directs the change to the Alter Tables panel.
3. After all the changes have been specified, enter ALTER on the Command line or click the **ALTER** keyword located at the top of the Alter Tables panel. Refer to number 3 in Figure 7-6.

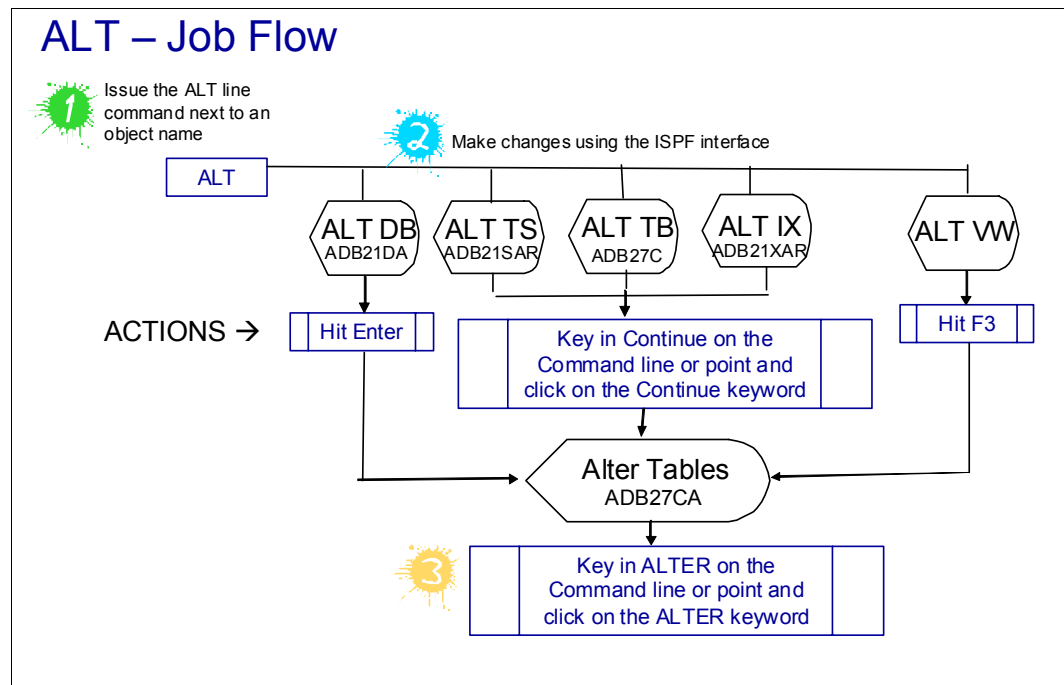


Figure 7-6 ALT Job Flow

4. If the “Show this panel prior to each use” field on the Alter Analysis Options panel has been set to YES, then the panel is displayed. After all the parameters have been set, press Enter to display the Build Analyze and Apply Job panel (Figure 7-7).

```

----- ALTER - Build Analyze and Apply Job
-----
Option ==>

Specify the following:

+
Worklist information:
  Worklist name . . . . . ALTER      (also used as middle qualifier in
DSNs)
  Prefix for data sets . . . DBA104

Data set information:
  PDS final qualifiers . . . JCL.CNTL
  Member name . . . . . ADBALTER
  Delete member name . . . ADBDELET (Optional job to delete work data
sets)

Options:
  Generate online . . . . . NO        (Yes/No)
  Generate one job . . . . . YES      (Yes/No)
  Member name or prefix . . APP12
  As work statement list . . YES      (Yes/No)
  Unload method . . . . . U          (Unload, Parallel unload, HPU)

Optional processes:
  Run CHECK DATA . . . . . NO        (Yes/No)

```

Figure 7-7 ALTER Build Analyze and Apply Job

5. Edit or modify the parameters on the Build Analyze and Apply Job panel. The information entered is used to determine how the change JCL is built and where it is stored.

After all the parameters on the Build Analyze and Apply panel have been updated, press Enter. Refer to number 5 in Figure 7-8.

6. The Data Set Existence Check panel opens and prompts the user to reuse the old data sets or to specify a new qualifier so that new data sets can be created. Refer to number 6 in Figure 7-8.

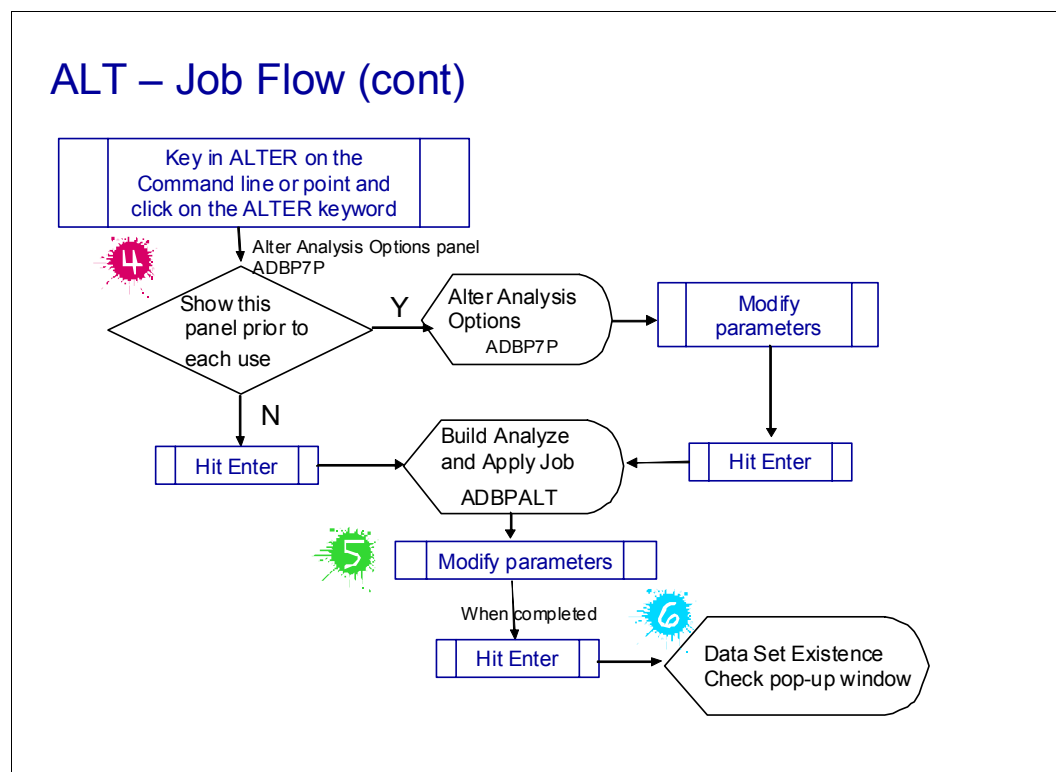


Figure 7-8 ALT Job Flow (continued)

7. If you opted to store the APPLY in a work statement list, you are prompted to supply the name of the work statement list. After the name has been supplied, press Enter. Refer to number 7 in Figure 7-9.
8. Two jobs, having the default names of ADBALTER and ADBDELET, are generated in a PDS that was specified on the Build Analyze and Apply parameter panel. Refer to number 8 in Figure 7-9.
 - a. ADBALTER consists of two parts, ANALYZE and APPLY.
 - b. ADBDELET is an optional set of JCL that can be used to delete underlying work data sets after a change has been successfully completed.

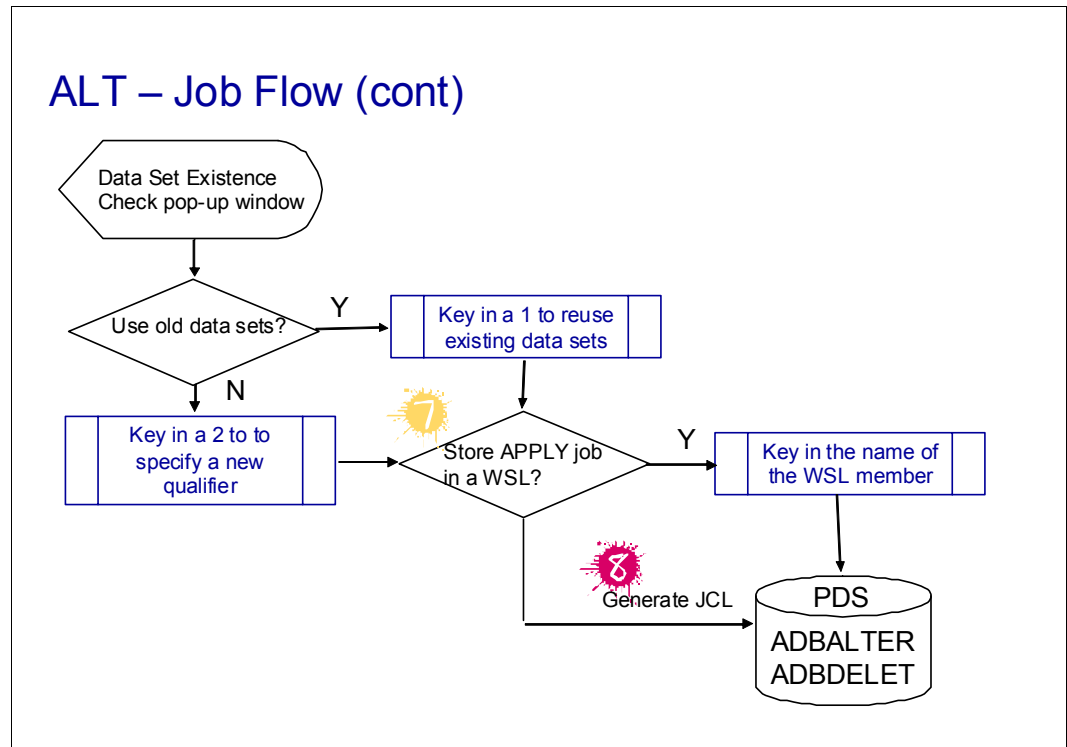


Figure 7-9 ALT Job Flow (continued)

9. Submit the ADBALTER JCL (Figure 7-10).
 - a. The ANALYZE step may be run online or in batch and determines what actions are needed to make the requested changes. If run in batch, it produces a report identifying the changes.
 - b. The APPLY step, based on the specifications in the Build, Analyze and Apply Job panel, generates the APPLY job either as a work statement list member in the work statement list library or an MVS JCL job stream in a PDS. Refer to number 9 in Figure 7-10.
 - c. To actually make or apply the change, you must either run the work statement list member online or in batch, or submit the MVS JCL APPLY job residing in the PDS. See number 10 in Figure 7-10.
10. After the change has successfully completed, you may choose to submit the ADBDELET job to delete the old work data set. Refer to number 11 in Figure 7-10.

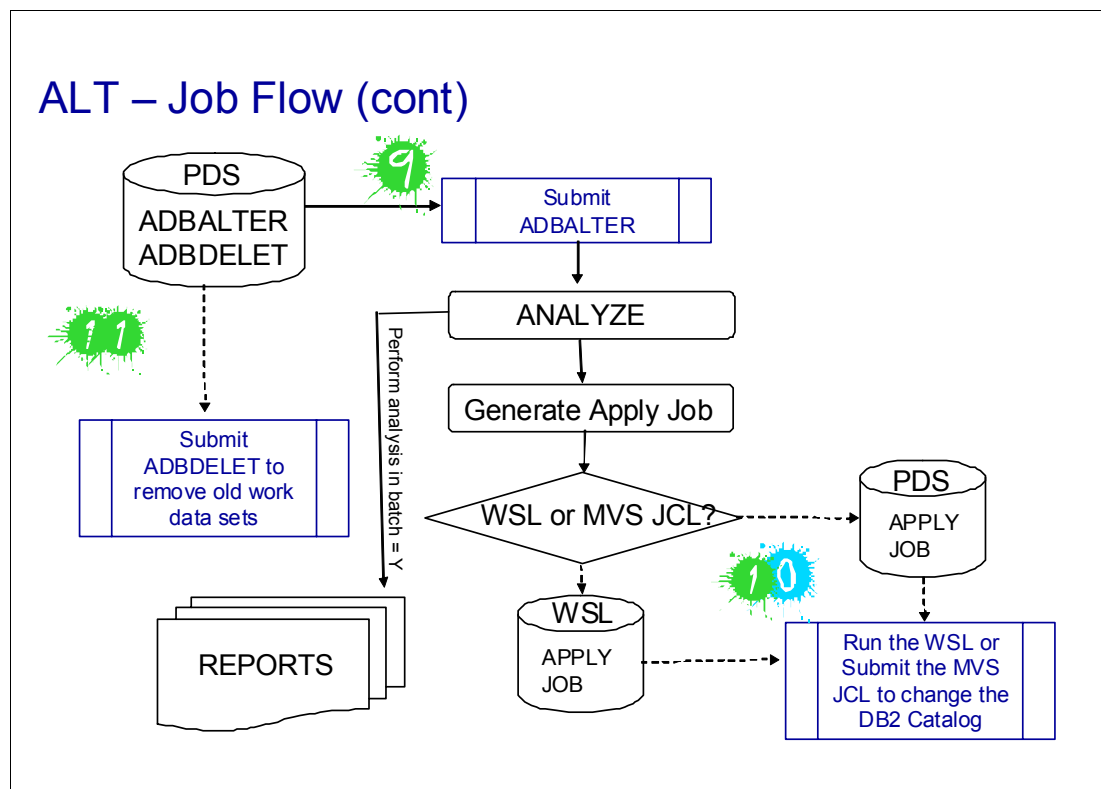


Figure 7-10 ALT Job Flow (continued)

Adding additional objects to a change

Change requests often consist of a set of changes against multiple objects or types of objects, making it desirable to incorporate all of the changes into one change definition. In previous releases, when using ALC, a user was able to add tables to a change session by entering the full schema name and a full or partial table name. With the advent of the ALT command, DB2 Administration Tool for z/OS V10 has provided the ability to include multiple tables and other objects using the Alter Tables panel.

To start a change request, the user may issue an ALT command next to a database, a table space, an index, or a view name. The change is inserted at the bottom of the Alter Tables panel.

To bring in other objects, the user, as in the past, must add one or more tables to the list of changed objects by entering the appropriate information on the Alter Tables panel.

There are two options for adding tables (Figure 7-11):

- Specify the name of the table explicitly by entering the full Schema and table name in the center of the Alter Tables panel under the heading “To add a table...”. The new table name is automatically appended to the list of changes at the bottom of the panel and has a NONE displayed under the Oper column on the far right.
- If you do not know the full name of the table, a list of tables may be generated by entering the full schema name of the table and a partial table name, with a question mark appended to the end, which acts as a wildcard character. An example is shown in Figure 7-11.

```

DB2 Admin ----- DBOB Alter Tables ----- Row 1 to 1 of 1
Command ==> Scroll ==> PAGE

Commands:
  ALTER - generate jobs  ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76  > (Table Schema)
Name . . . . td76?  > (Table Name. ? to look up)

Object  Object
Sel Qual  Name          T DB Name  TS Name    RI RI  FK  Chg
*         *             * *          *          Rls Add Add Rqd Oper
*         *             * *          *          * *  *  *  *  *

-----> -----> -----
TDTEAM76 TD76TS02      S TDTEAM76 TD76TS02      NA  NA      MODIFY
***** END OF DB2 DATA *****

```

Figure 7-11 Alter Tables: Adding a table using wildcards

After the list of tables opens, the user may select one or more tables to be included in the change by entering a plus (+) sign next to the name of one or more of the tables. An example of the output resulting from using the question mark as a wildcard can be seen in Figure 7-12.

```
DB2 Admin ----- DSNT Tables, Views, and Aliases ----- Row 1 from 14
Command ==>
Select by typing '+'
Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands
```

Sel	Name	Schema	T DB Name	TS Name	Cols	Rows	Chks
*	*	*	* *	*	*	*	*
* -----> -----							
+	TD76TB01_DEPT	TEAM76	T TDTEAM76	TD76TS01	5	-1	0
+	TD76TB02_EMP	TEAM76	T TDTEAM76	TD76TS02	14	-1	2
	TD76TB03_ACT	TEAM76	T TDTEAM76	TD76TS03	3	-1	0
	TD76TB04_PROJ	TEAM76	T TDTEAM76	TD76TS04	8	-1	0
	TD76TB05_PROJACT	TEAM76	T TDTEAM76	TD76TS05	5	-1	0
	TD76TB06_EMPPROJAC	TEAM76	T TDTEAM76	TD76TS06	6	-1	0
	TD76TB07_EACT	TEAM76	T TDTEAM76	TD76TS07	5	-1	0
	TD76TB08_EDEPT	TEAM76	T TDTEAM76	TD76TS08	7	-1	0
	TD76TB09_EEPA	TEAM76	T TDTEAM76	TD76TS09	8	-1	0
	TD76TB10_EPROJ	TEAM76	T TDTEAM76	TD76TS10	10	-1	0
	TD76TB11_EPROJACT	TEAM76	T TDTEAM76	TD76TS11	7	-1	0
	TD76TB12_EEMP	TEAM76	T TDTEAM76	TD76TS12	16	-1	0

Figure 7-12 Select tables using the plus sign

After the objects have been selected, a row for each table is inserted at the bottom of the ALTER Tables panel (Figure 7-13).

```
DB2 Admin ----- DB0B Alter Tables ----- Row 1 to 3 of 3
Command ==>                                     Scroll ==> PAGE

Commands:
  ALTER - generate jobs  ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76  > (Table Schema)
Name . . . . . > (Table Name. ? to look up)

  Object  Object
Sel Qual  Name          T DB Name  TS Name  RI RI  FK  Chg
*          *            * *          *      * *  *  *  *
-----> -----> -----> -----> -----> -----> ----->
  TEAM76  TD76TB02_EMP    T TDTEAM76 TD76TS02    0 NA  NA  NONE
  TEAM76  TD76TB01_DEPT  T TDTEAM76 TD76TS01    0 NA  NA  NONE
  TDTEAM76 TD76TS02      S TDTEAM76 TD76TS02    NA NA  MODIFY
***** END OF DB2 DATA *****
```

Figure 7-13 Alter Tables panel with newly added objects

To alter the table definition, enter the A (Alter Object) line command next to the name of the table (Figure 7-14).

```
DB2 Admin ----- DSNT Alter Tables ----- Row 1 to 3 of 3
Command ==>                                     Scroll ==> CSR

Commands:
  ALTER - generate jobs  ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76  >          (Table Schema)
Name . . . .          > (Table Name. ? to look up)

      Object  Object
Sel Qual  Name          T DB Name  TS Name    RI RI  FK  Chg
      *      *          * *        *          *  *  *  *  *
-----> -----> -----> -----> -----> -----> -----> ----->
a  TEAM76  TD76TB02_EMP  T TDTEAM76 TD76TS02    0 NA  NA      NONE
    TEAM76  TD76TB01_DEPT T TDTEAM76 TD76TS01    0 NA  NA      NONE
    TDTEAM76 TD76TS02    S TDTEAM76 TD76TS02    NA NA      MODIFY
***** END OF DB2 DATA *****
```

Figure 7-14 Alter a table using the A command

```
DB2 Admin ----- DSNT Alter Tables ----- Row 1 to 3 of 3
Command ==> Scroll ==> CSR

Commands:
ALTER - generate jobs   ADDFK - Add Foreign Key-affected tables
OPTIONS - Change alter options

Line commands:
A - Alter Object D - Delete S - Select Object REL - Alter related
FK - Add Foreign Key-affected tables E - Edit DDL
RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76 > (Table Schema)
Name . . . . > (Table Name. ? to look up)


Sel Qual      Object      Object      T DB Name TS Name      RI Rel's FK Add Rqd Oper
* *          *           *           * *         *       * *  *    *
-----> -----> - -----> -----> -----> -----> ----->
TEAM76 TD76TB02_EMP T TDTEAM76 TD76TS02 O NA NA NONE
rEL TEAM76 TD76TB01_DEPT T TDTEAM76 TD76TS01 O NA NA NONE
TDTEAM76 TD76TS02 S TDTEAM76 TD76TS02 NA NA MODIFY
***** END OF DB2 DATA *****
```

Press Enter, and the list of related objects is displayed on the Alter Related panel (Figure 7-16).

```
DB2 Admin ----- DSNT Alter Related : TEAM76.TD76TB01_DEPT -- Row 1 to 9 of 9
Command ==>                                         Scroll ==> CSR
```

Commands: ADDINDEX ADDFK
Line commands: A - Alter object S - Show object R - Recreate view

S	Type	Qual	Name	Dep.	Views	Rel	Name	Note
*		*	*		*	*		*

	DATABASE		TDTEAM76		0			
	INDEX	TEAM76	TD76XA01		0			
	INDEX	TEAM76	TD76XB01		0			
	INDEX	TEAM76	TD76XC01		0			
	TABLESPACE	TDTEAM76	TD76TS01		0			
	VIEW	TEAM76	VWDEPMG1		2			
	VIEW	TEAM76	VWDEPT		0			
	VIEW	TEAM76	VWEMPDPT1		0			
	VIEW	TEAM76	VWHDEPT		0			
***** END OF DB2 DATA *****								

Figure 7-16 Alter related objects

To change one of the related objects, the user issues the A (Alter Object) line command next to the object name to be altered. In Figure 7-17, an “A” has been issued next to the table space name.

```
DB2 Admin ----- DSNT Alter Related : TEAM76.TD76TB01_DEPT -- Row 1 to 9 of 9
Command ==>                                         Scroll ==> CSR
```

Commands: ADDINDEX ADDFK
Line commands: A - Alter object S - Show object R - Recreate view

S	Type	Qual	Name	Dep.	Views	Rel	Name	Note
*		*	*		*	*		*

	DATABASE		TDTEAM76		0			
	INDEX	TEAM76	TD76XA01		0			
	INDEX	TEAM76	TD76XB01		0			
	INDEX	TEAM76	TD76XC01		0			
a	TABLESPACE	TDTEAM76	TD76TS01		0			
	VIEW	TEAM76	VWDEPMG1		2			
	VIEW	TEAM76	VWDEPT		0			
	VIEW	TEAM76	VWEMPDPT1		0			
	VIEW	TEAM76	VWHDEPT		0			
***** END OF DB2 DATA *****								

Figure 7-17 Alter Related: Alter a table space

The change panel that opens depends on the type of object being explained. In this case, the Redefine Table Space panel opens (Figure 7-18).

When adding objects in this manner, it is possible that the change request could be accomplished by using an ALTER statement. Regardless, the ISPF panel displayed when issuing the A (Alter) command next to the object name includes all parameters. The DB2 Administration Tool utilizes its “Smart Alter” facility and only generates an ALTER if that is all that is needed to make the change.

```

DB2 Admin ----- DSNT Redefine Table Space ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Commands: CONTINUE ORIGINAL MAKEPBG MAKEPBR
Line commands: S - Split part R - Remove part O - Original data
               C - Clear data
CREATE TABLESPACE: TD76TS01 IN TDTEAM76

Numparts . . . . . 0                                LOB . . . . . NO
Define . . . . . YES                                DSSIZE . . . . .
Member Cluster . . NO                                SEGSIZE . . . . 4    CCSID . . . . EBCDIC
Buffer Pool . . . . BPO                                Close Rule . . NO    Max Rows . . 255
Lock Size . . . . . PAGE                                Lock Part . . . NO    Lock Max . . SYSTEM
Max Partitions . . 0

S Part      Pqty      Sqty Page Free Compr R M T VCAT      Stogroup
GBPCache
- - - - -
0          -1          -1    0    5 NO    N Y I DSNTCAT DSN8G100 CHANGED
***** END OF DB2 DATA *****

```

Figure 7-18 Redefine a table space

The changed object is flagged as having a modification pending (Figure 7-19.)

```
DB2 Admin ----- DSNT Alter Related : TEAM76.TD76TB01_DEPT -- Row 1 to 9 of 9
Command ==>                                         Scroll ==> CSR
```

Commands: ADDINDEX ADDFK
Line commands: A - Alter object S - Show object R - Recreate view

S	Type	Qual	Name	Dep.	Views	Rel	Name	Note
*		*	*	* *				*
-	-----	-----	-----	-----	-----	-----	-----	-----
	DATABASE		TDTEAM76	0				
	INDEX	TEAM76	TD76XA01	0				
	INDEX	TEAM76	TD76XB01	0				
	INDEX	TEAM76	TD76XC01	0				
*	TABLESPACE	TDTEAM76	TD76TS01	0				Modification pending
	VIEW	TEAM76	VWDEPMG1	2				
	VIEW	TEAM76	VWDEPT	0				
	VIEW	TEAM76	VWEMPDP1	0				
	VIEW	TEAM76	VWHDEPT	0				
***** END OF DB2 DATA*****								

Figure 7-19 Modified Alter Related panel

To return to the Alter Tables panel, the user must press F3. All of the new changes are found at the bottom of the pane (Figure 7-20).

```
DB2 Admin ----- DB0B Alter Tables ----- Row 1 to 4 of 4
Command ==>                                         Scroll ==> PAGE

Commands:
  ALTER - generate jobs  ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options

Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76  > (Table Schema)
Name . . . . . > (Table Name. ? to look up)

  Object      Object
Sel Qual      Name
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *           *
  *          
```

Figure 7-20 Modified Alter Tables Panel

When all of the objects included in the change request have been modified, enter ALTER on the Command line or click the **ALTER** keyword at the top of the Alter Tables panel. If the “Show this panel prior to each use” option is set to YES on the ALTER Analysis Options panel, then that panel is displayed first. If so, press Enter to open the Build Analyze and Apply Job panel. If the “Show this panel prior to each use” option is set to NO, then the Build Analyze and Apply Jobs panel opens (Figure 7-21).

----- ALTER - Build Analyze and Apply Job -----

Option ==>

Specify the following:

More: +

Worklist information:

Worklist name ALTER (also used as middle qualifier in DSNs)

Prefix for data sets . . . DBA104

Data set information:

PDS final qualifiers . . . JCL.CNTL

Member name ADBALTER

Delete member name . . . ADBDELET (Optional job to delete work data sets)

Options:

Generate online NO (Yes/No)

Generate one job YES (Yes/No)

Member name or prefix . . APP12

As work statement list . . YES (Yes/No)

Unload method U (Unload, Parallel unload, HPU)

Optional processes:

Run CHECK DATA NO (Yes/No)

Figure 7-21 Build Analyze and Apply Jobs

Build, Analyze, and Apply Parameter panel

The Build, Analyze, and Apply Parameter panel is used to specify how and where the apply job is to be generated. After the parameters on the panel are entered, two jobs are generated in a PDS: ADBALTER and ADBDELET. When executed, ADBALTER generates the JCL for the application of the change. ADBDELET is optional and can be used to delete old work data sets after a change has been successfully completed. Descriptions of the various options on the Build Analyze and Apply panel are described in Table 7-3.

Table 7-3 Build, Analyze, and Apply Parameter panel options

Description	Option	Values or defaults	Comments
Worklist information	Worklist name	Any eight character name	The value entered is used as a middle qualifier in work data sets and is also used as the default work statement list member name.
	Prefix for data sets	17 characters	The high level qualifier for data set names that are generated during the change process.

Description	Option	Values or defaults	Comments
Data set	PDS low level qualifiers	48 characters	The PDS suffix for the JCL library where the ADBALTER and ADBDELET jobs are stored.
	Alter JCL member name	ADBALTER	This is the member name of the JCL that, when submitted, performs the ANALYZE and generates the APPLY into a work statement list or as MVS JCL in a PDS. The default member name is ADBALTER, but any name can be used as long as it is not greater than eight characters in length
	Delete JCL member name	ADBDELET	An optional job that is used to delete work data sets after the change has completed. The default member name is ADBDELET, but any name can be used as long as it is not greater than eight characters in length.
Options	Generate online	YES or NO	Perform the analysis online or in batch. If done online, your terminal is tied up during the ANALYZE, as the process runs in foreground and there are no reports generated. If done in batch, ANALYZE is a step in the ADBALTER job. Reports are produced indicating what actions are going to be taken to complete the change.
	Generate one job	YES or NO	You can generate one or multiple apply jobs. This parameter is only applicable if are generating the APPLY jobs into a PDS. If does not apply if you are storing them in a WSL member. If you generate multiple jobs, the job names that can run in parallel have the same prefix. The issue with generating multiple jobs is that you have to submit each one separately
	Member name or prefix for the generated jobs	APPLY	The member name or prefix for the generated apply job if and when Generate one job = YES and the output is being routed to a PDS.
	As work statement list (WSL)	YES or NO	If = YES, store the APPLY in a work statement list; If = NO, store the APPLY in a JCL member in a PDS file.
	Unload method	Unload / Parallel / HPU	If routing the jobs to a WSL, the unload method must be U.

Description	Option	Values or defaults	Comments
Optional processes - to be executed after a change has successfully completed	Run CHECK DATA	YES or NO	Run CHECK data to avoid check pending.
	Run COPY	AFTER RELOAD / ALTER / BOTH / NONE	Specify when you want to make an image copy.
	Run REORG / REBUILD	MANDATORY / ALL RELEVANT / NONE	Specify when you want to run a REORG or REBUILD: <ul style="list-style-type: none"> ► Mandatory: Generates the necessary REORGs to remove the REORG pending condition. ► All relevant: Generates all the necessary REORGs needed to fully implement the effects of changes.
	Run RUNSTATS	AFTER RELOAD / ALTER / BOTH / NONE	Specify when to run RUNSTATS.
	Run REBIND	YES or NO	
Utility Control Options	Use templates	YES or NO	Templates provide the means of overriding the default DSN for DB2 Admin processes.
	Use utility options	YES or NO	Use the utility options specified by the user rather than the defaults delivered with the DB2 Administration Tool skeletons.

7.2 Using ALT

The *IBM DB2 Administration Tool for z/OS Version 10 Release 1 User's Guide and Reference*, SC19-3033 contains several examples of using ALT:

- RENAME a database.
- Redefine a table space.
- Alter a table.
- Alter multiple tables.
- Alter different types of objects.

For the purposes of this publication, we have included another example of adding different objects to a change definition.

7.2.1 Example of adding multiple object types to a change

Our change request contains the following items:

1. EMPLOYEE table
 - a. Add CELLPHONE to the EMP table.
 - b. Add PHONE and CELLPHONE to the VWEMPDPDT1 view.

2. DEPT table
 - a. Add a new column BUILDING CHAR 5 to the end of the DEPT table.
 - b. Lengthen the LOCATION field to 25 characters.
3. EMPPROJECT table space
 - a. Change LOG YES to LOG NO.
4. INCLUDE column in the EMPPROJECT index (DB2 10 for z/OS feature)
 - a. Include EMPTIME to unique index, which in our example is TD76XA06.

Making the change

Enter ALT next to the EMP table name (Figure 7-22). The full name of the table we are using for the document is TD76TB02_EMP.

```
DB2 Admin ----- DB0B Tables, Views, and Aliases ----- Row 1 from 14
Command ==> Scroll ==> PAGE

Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands
```

Sel	Name	Schema	T DB Name	TS Name	Cols	Rows	Chks
	*	*	* *	*	*	*	*
----->-----							
	TD76TB01_DEPT	TEAM76	T TDTEAM76	TD76TS01	5	-1	0
alt	TD76TB02_EMP	TEAM76	T TDTEAM76	TD76TS02	14	-1	2
	TD76TB03_ACT	TEAM76	T TDTEAM76	TD76TS03	3	-1	0
	TD76TB04_PROJ	TEAM76	T TDTEAM76	TD76TS04	8	-1	0
	TD76TB05_PROJACT	TEAM76	T TDTEAM76	TD76TS05	5	-1	0
	TD76TB06_EMPPROJAC	TEAM76	T TDTEAM76	TD76TS06	6	-1	0
	TD76TB07_EACT	TEAM76	T TDTEAM76	TD76TS07	5	-1	0
	TD76TB08_EDEPT	TEAM76	T TDTEAM76	TD76TS08	7	-1	0
	TD76TB09_EEPA	TEAM76	T TDTEAM76	TD76TS09	8	-1	0
	TD76TB10_EPROJ	TEAM76	T TDTEAM76	TD76TS10	10	-1	0
	TD76TB11_EPROJACT	TEAM76	T TDTEAM76	TD76TS11	7	-1	0
	TD76TB12_EEMP	TEAM76	T TDTEAM76	TD76TS12	16	-1	0

Figure 7-22 ALT line command example

Add CELLPHONE to the EMP table (Figure 7-23).

DB2 Admin ----- DB0B ALTER Table ----- Row 1 to 7 of 15
Command ==> Scroll ==> PAGE

New schema . . TEAM76 >
Old schema: TEAM76

New name . . . TD76TB02_EMP >
Old name : TD76TB02_EMP

Volatile . . . NO
New DB . . TDTEAM76

Rows per page: 36
Partitions: 5
New TS . . TD76TS02

Commands: CONTINUE PRIMKEY ADD REL ALTPART
Line commands:
I - Insert U - Update D - Delete R - Repeat LAB - Label COM - Comment
M - Move A - After B - Before X - Index RES - Reset update

Old Operation

Seq	Column Name	Col No	Col Type	Length	Scale	N	D	Col No	Type
*		* *		*		*	*	*	*
----->-----									
	EMPNO	1	CHAR	6	0	N	N	1	
	FIRSTNME	2	VARCHAR	12	0	N	N	2	
	MIDINIT	3	CHAR	1	0	N	N	3	
	LASTNAME	4	VARCHAR	15	0	N	N	4	
	WORKDEPT	5	CHAR	3	0	Y	Y	5	
*	PHONENO	6	CHAR	4	0	Y	Y	6	
*	CELLPHONE	7	CHAR	4	0	Y	Y	0	INSERT

Figure 7-23 Insert a new column named CELLPHONE

Our change request includes altering the VWEMPDP1 view to add both the PHONENO and CELLPHONE columns to the definition of the view. To modify a related view, you must use the REL command.

We began our change session by entering ALT next to the name of the EMP table. When the ALTER Table panel (ADB27C) opens, a valid REL command is listed at the top of the panel. We see later in our change definition that the REL command for tables added to the change scenario is only available on the Alter Tables panel (ADB27CA).

Enter REL on the Command line of the ALTER Tables panel (ADB27C) (Figure 7-24).

```
ADB27C in ----- DB0B ALTER Table ----- Row 1 to 7 of 15
Command ==> rel                               Scroll ==> PAGE

New schema . . TEAM76 >                      Old schema: TEAM76
New name . . . TD76TB02_EMP >                Old name : TD76TB02_EMP
Volatile . . . NO                             New DB . . TDTEAM76
Rows per page: 36          Partitions: 5      New TS . . TD76TS02

Commands: CONTINUE PRIMKEY ADD REL ALTPART
Line commands:
I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
M - Move    A - After   B - Before  X - Index   RES - Reset update
                                           Old Operation
Sel Column Name      Col No Col Type      Length  Scale N D Col No Type
*                   * *
----->-----
EMPNO                1 CHAR                6      0 N N      1
FIRSTNME             2 VARCHAR            12      0 N N      2
MIDINIT              3 CHAR                 1      0 N N      3
LASTNAME             4 VARCHAR            15      0 N N      4
WORKDEPT             5 CHAR                 3      0 Y Y      5
* PHONENO            6 CHAR                 4      0 Y Y      6
* CELLPHONE          7 CHAR                 4      0 Y Y      0 INSERT
```

Figure 7-24 REL command to display related objects

To add the two columns to the view, enter the A (Alter Object) line command next to the VWEMDPT view name (Figure 7-25).

```
ADB27CR n ----- DB0B Alter Related : TEAM76.TD76TB02_EMP - Row 1 to 14 of 14
Command ==>                                     Scroll ==> PAGE
```

Commands: ADDINDEX ADDFK
Line commands: A - Alter object S - Show object R - Recreate view

S	Type	Qual	Name	Dep.	Views	Rel	Name	Note
*		*	*		*	*		*

	DATABASE		TDTEAM76		0			
	INDEX	TEAM76	TD76XA02		0			
	INDEX	TEAM76	TD76XB02		0			
	TABLESPACE	TDTEAM76	TD76TS02		0			
	VIEW	TEAM76	TD76ASTRDE2		0			
	VIEW	TEAM76	TD76DEPMG1		2			
	VIEW	TEAM76	TD76EMPDPT1		0			
	VIEW	TEAM76	VWASTRDE2		0			
	VIEW	TEAM76	VWDEPMG1		2			
	VIEW	TEAM76	VWEMP		0			
a	VIEW	TEAM76	VWEMPDPT1		0			
	VIEW	TEAM76	VWEMPLP		0			
	VIEW	TEAM76	VWPROJRE1		3			
	VIEW	TEAM76	VWSTAFAC2		0			

Figure 7-25 Example of issuing the Alter command

DB2 Administration Tool provides the ability to display all of the column names in a view or to retain view definitions where the view is defined by a SELECT *.

For the purposes of this scenario, we are going to select option 2 to create views without a column list (Figure 7-26).

```
DB2 Admin -- DB0B Alter Related - View Column List ----- 13:26
```

Views created without a column list may again be created without a column list. As an alternative, column names can be provided from the result table.
Note: Views which had a column list are not affected.

Select a choice

- 1. Create a column list from the result table**
- Create such views without a column list

Figure 7-26 Alter Related - View Column List

Enter the column names in the view text (Figure 7-27).

```

000001 SET CURRENT SCHEMA='ADMR1';
000002 SET CURRENT PATH = "TEAM76" ;
000003 CREATE VIEW TEAM76.VWEMPDPPT1(DEPTNO, DEPTNAME, EMPNO, FRSTINIT,
000004     MIDINIT, LASTNAME, WORKDEPT. PHONE, CELLPHONE) AS
000005     SELECT ALL TEAM76.TD76TB01_DEPT.DEPTNO,
000006         TEAM76.TD76TB01_DEPT.DEPTNAME, TEAM76.TD76TB02_EMP.EMPNO,
000007         SUBSTR(Team76.TD76TB02_EMP.FIRSTNME, 1, 1),
000008         TEAM76.TD76TB02_EMP.MIDINIT, TEAM76.TD76TB02_EMP.LASTNAME,
000009         TEAM76.TD76TB02_EMP.WORKDEPT, TEAM76.TD76TB02_EMP.PHONE,
000010         TEAM76.TD76TB02_EMP.CELLPHONE
000011     FROM TEAM76.TD76TB01_DEPT
000012     RIGHT OUTER JOIN
000013     TEAM76.TD76TB02_EMP
000014     ON TEAM76.TD76TB02_EMP.WORKDEPT = TEAM76.TD76TB01_DEPT.
000015     DEPTNO ;
000016 COMMIT;

```

Figure 7-27 Add the columns to the view

Return to the Alter Related Table panel by pressing F3. Note the message to the right of the view name indicating that there is a modification pending for this object (Figure 7-28).

```
DB2 Admin ----- DB0B Alter Related : TEAM76.TD76TB02_EMP - Row 1 to 14 of 14
Command ==>
Scroll ==> PAGE

Commands: ADDINDEX  ADDFK
Line commands: A - Alter object  S - Show object  R - Recreate view
```

S Type	Qual	Name	Dep.	Views	Rel	Name	Note
*	*	*		*	*		*

	DATABASE	TDTEAM76		0			
	INDEX	TEAM76 TD76XA02		0			
	INDEX	TEAM76 TD76XB02		0			
	TABLESPACE	TDTEAM76 TD76TS02		0			
	VIEW	TEAM76 TD76ASTRDE2		0			
	VIEW	TEAM76 TD76DEPMG1		2			
	VIEW	TEAM76 TD76EMPDPPT1		0			
	VIEW	TEAM76 VWASTRDE2		0			
	VIEW	TEAM76 VWDEPMG1		2			
	VIEW	TEAM76 VWEMP		0			
*	VIEW	TEAM76 VWEMPDPPT1		0			Modification pending
	VIEW	TEAM76 VWEMPLP		0			
	VIEW	TEAM76 VWPROJRE1		3			
	VIEW	TEAM76 VWSTAFAC2		0			

Figure 7-28 Modification pending


```
DB2 Admin ----- DB0B Alter Tables ----- Row 1 to 2 of 2
Command ==> Scroll ==> PAGE
```

Commands:

- ALTER - generate jobs ADDFK - Add Foreign Key-affected tables
- OPTIONS - Change alter options

Line commands:

- A - Alter Object D - Delete S - Select Object REL - Alter related
- FK - Add Foreign Key-affected tables E - Edit DDL
- RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter

Schema . . . TEAM76 > (Table Schema)

Name . . . td76? > (Table Name. ? to look up)

Object Sel Qual	Object Name	T	DB	Name	TS	Name	RI	RI	FK	Chg
*	*	*	*	*	*	*	Rel's	Add	Add	Rqd Oper
*	*	*	*	*	*	*	*	*	*	*
----	----->	----->	-----	-----	-----	-----	-----	-----	-----	-----
TEAM76	VWEMPDPT1	V					NA	NA		MODIFY
*EL TEAM76	TD76TB02_EMP	T	TDTEAM76	TD76TS02			O NA	NA		MODIFY
***** END OF DB2 DATA *****										

A list of tables matching the wildcard prefix opens. Select the DEPT and the EMP PROJACT table using the plus sign (Figure 7-31).

```

DB2 Admin ----- DB0B Tables, Views, and Aliases ----- Row 1 from 14
Command ==>
Select by typing '+'
Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands

```

Sel	Name	Schema	T DB Name	TS Name	Cols	Rows	Chks
	*	*	* *	*	*	*	*
----->							
+	TD76TB01_DEPT	TEAM76	T	TDTEAM76	TD76TS01	5	-1 0
	TD76TB02_EMP	TEAM76	T	TDTEAM76	TD76TS02	14	-1 2
	TD76TB03_ACT	TEAM76	T	TDTEAM76	TD76TS03	3	-1 0
	TD76TB04_PROJ	TEAM76	T	TDTEAM76	TD76TS04	8	-1 0
	TD76TB05_PROJACT	TEAM76	T	TDTEAM76	TD76TS05	5	-1 0
+	TD76TB06_EMP PROJACT	TEAM76	T	TDTEAM76	TD76TS06	6	-1 0
	TD76TB07_EACT	TEAM76	T	TDTEAM76	TD76TS07	5	-1 0
	TD76TB08_EDEPT	TEAM76	T	TDTEAM76	TD76TS08	7	-1 0
	TD76TB09_EEPA	TEAM76	T	TDTEAM76	TD76TS09	8	-1 0
	TD76TB10_EPROJ	TEAM76	T	TDTEAM76	TD76TS10	10	-1 0
	TD76TB11_EPROJACT	TEAM76	T	TDTEAM76	TD76TS11	7	-1 0
	TD76TB12_EEMP	TEAM76	T	TDTEAM76	TD76TS12	16	-1 0

Figure 7-31 List of tables matching the wildcard criteria

Be sure to press Enter to include the two tables. The + signs converts to asterisks. If you press F3, the selection is not made. After the tables have been selected and you have pressed Enter, return to the Alter Tables panel (ADB27CA) by pressing F3.

[illegible]

To change the definition of the DEPT table (TD76TB01_DEPT), enter an A (Alter Object) line command next to the DEPT table name (Figure 7-33).

```

DB2 Admin ----- DB0B Alter Tables ----- Row 1 to 4 of 4
Command ==>                                     Scroll ==> PAGE

Commands:
  ALTER - generate jobs  ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76  > (Table Schema)
Name . . . . . > (Table Name. ? to look up)

      Object  Object
Sel Qual  Name          T DB Name  TS Name      RI RI  FK  Chg
*         *            * *          *          Rels Add Add Rqd Oper
*         *            * *          *          * *  *  *  *
-----> -----> - -----> -----> -----> ----->
      TEAM76  TD76TB06_EMPPROJAC T TDTEAM76 TD76TS06      0 NA  NA      NONE
a  TEAM76  TD76TB01_DEPT      T TDTEAM76 TD76TS01      0 NA  NA      NONE
      TEAM76  VWEMPDPT1          V              NA  NA      MODIFY
*EL TEAM76  TD76TB02_EMP          T TDTEAM76 TD76TS02      0 NA  NA      MODIFY
***** END OF DB2 DATA *****

```

Figure 7-33 Alter Tables

Originally this change request began when the ALT line command was issued against the EMP table. As additional objects are added to the list of changes, the A line command is used to alter those objects.

To make the requested changes to the DEPT table, issue the A line command next to the table name. Add the new BUILDING column for a CHAR length of 5 and modify the LOCATION column to a length of 25.

Notice the messages under the Operation Type column heading on the far right side of the panel. BUILDING is flagged as an INSERT and the LOCATION column is an UPDATE (Figure 7-34).

```

DB2 Admin ----- DB0B ALTER Table ----- Row 1 to 6 of 6
Command ==>                                     Scroll ==> PAGE
View DDL update reqd.
New schema . . TEAM76      >                      Old schema: TEAM76
New name . . . TD76TB01_DEPT      >                Old name  : TD76TB01_DEPT
Volatile . . . NO                      New DB   . . TDTEAM76
Rows per page: 53          Partitions: 0          New TS   . . TD76TS01

Commands: CONTINUE PRIMKEY
Line commands:
I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
M - Move    A - After   B - Before  X - Index   RES - Reset update
                                           Old Operation
Sel Column Name          Col No Col Type      Length  Scale N D Col No Type
*                        * *
-----> -----
DEPTNO                    1 CHAR              3        0 N N      1
DEPTNAME                  2 VARCHAR          36        0 N N      2
MGRNO                     3 CHAR              6        0 Y Y      3
ADMRDEPT                  4 CHAR              3        0 N N      4
*  LOCATION                5 CHAR             25        0 Y Y      5 UPDATE
  BUILDING                 6 CHAR              5        0 Y Y      0 INSERT
***** END OF DB2 DATA *****

```

Figure 7-34 ALTER Table

Notice at the top of the ALTER Table panel that there is a message reminding the user that View DDL update reqd, but REL is no longer a valid line command on the ALTER Table panel (ADB27C). To modify related objects on the recently added tables, the user must return to the Alter Table panel (ADB27CA) and issue REL as a line command next to the name of the table.

Return to the Alter Tables panel by either entering CONTINUE on the Command line or click the **CONTINUE** keyword at the top of the ALTER Table panel. If the Alter Related Objects panel opens, select either option 1 or 2, as both take you to the Alter Tables panel (Figure 7-35).

```

DB2 Admin -- DBOB Alter Related Objects ----- 14:13

Table or column name changes have occurred which may require
editing of related objects.
Such objects are flagged with 'E' in the 'CHG Rqd' column.

Select a choice
1. Continue creating ALTER JCL
2. Return without building JCL

```

Figure 7-35 Alter Related Objects

Even though our change request did not include any updates to any objects related to the DEPT table, DB2 Administration Tool informs us that the changes we just made to the DEPT table have somehow impacted other objects in our change definition, and that these objects are now flagged with an E (Figure 7-36).

```

DB2 Admin ----- DBOB Alter Tables ----- Row 1 to 4 of 4
Command ==>                                     Scroll ==> PAGE

Commands:
  ALTER - generate jobs  ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76  >          (Table Schema)
Name . . . .        > (Table Name. ? to look up)

      Object  Object
Sel Qual  Name          T DB Name  TS Name  RI RI  FK  Chg
      *      *          *  *      *      *  *  *  *  *
-----> -----> - -----

EsxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxN
e Table or column names were changed. Related objects within this list which e
e need to be edited have the 'CHG Rqd' column set to 'E'.                      e
DsxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxM
***** END OF DB2 DATA *****

```

Figure 7-36 Alter Tables: Message regarding a CHG Rqd

Press Enter to clear the message at the bottom of the Alter Tables panel (Figure 7-37).

```
DB2 Admin ----- DB0B Alter Tables ----- Row 1 to 4 of 4
Command ==>                                     Scroll ==> PAGE

Commands:
  ALTER - generate jobs  ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76  > (Table Schema)
Name . . . . . > (Table Name. ? to look up)

      Object  Object
Sel Qual  Name          T DB Name  TS Name  RI RI  FK  Chg
*         *            * *          *         Rels Add Add Rqd Oper
*         *            * *          *         * *  *  *  *
-----> -----> -----> -----> -----> -----> ----->
*   TEAM76  TD76TB06_EMPPROJAC T TDTEAM76 TD76TS06    0 NA  NA      NONE
*   TEAM76  TD76TB01_DEPT      T TDTEAM76 TD76TS01    0 NA  NA      MODIFY
*   TEAM76  VWEMPDPT1          V              NA NA E MODIFY
*EL TEAM76  TD76TB02_EMP      T TDTEAM76 TD76TS02    0 NA  NA      MODIFY
***** END OF DB2 DATA *****
```

Figure 7-37 Alter Tables

The DEPT table has a MODIFY under the Oper column heading name.

The VWEMPDPT1 view has been flagged with an E under the Chg Rqd column heading. This flag is a result of the change we just made to the DEPT table and serves as a reminder that the view needs to be updated one more time before the change can be executed.

Because other changes in the request may also have an impact on the view, we do this update after all of the changes have been made.

```
DB2 Admin ----- DB0B Alter Tables ----- Row 1 to 4 of 4
Command ==> Scroll ==> PAGE

Commands:
ALTER - generate jobs   ADDFK - Add Foreign Key-affected tables
OPTIONS - Change alter options

Line commands:
A - Alter Object D - Delete S - Select Object REL - Alter related
FK - Add Foreign Key-affected tables E - Edit DDL
RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76 > (Table Schema)
Name . . . . > (Table Name. ? to look up)


      Object    Object                                RI RI  FK  Chg
Sel Qual       Name          T DB Name TS Name  Rels Add Add Rqd Oper
  *         *              * *           *            * *  *  *  *
--- ---->----->----->----->----->----->----->----->----->
rel TEAM76     TD76TB06_EMPPROJAC T TDTEAM76 TD76TS06      0 NA  NA      NONE
*   TEAM76     TD76TB01_DEPT      T TDTEAM76 TD76TS01      0 NA  NA      MODIFY
*   TEAM76     VWEMPDPT1          V                      NA  NA  E      MODIFY
*EL TEAM76     TD76TB02_EMP        T TDTEAM76 TD76TS02      0 NA  NA      MODIFY
***** END OF DB2 DATA *****
```

The Alter Related panel opens. Enter an A (Alter Object) next to the name of the table space (Figure 7-39).

```
DB2 Admin ----- DB0B Alter Related : TEAM76.TD76TB06_EMPPROJ Row 1 to 7 of 7
Command ==> Scroll ==> PAGE

Commands: ADDINDEX  ADDFK
Line commands: A - Alter object  S - Show object  R - Recreate view

S Type           Qual      Name              Dep.
*              *        *              Views Rel Name Note
*              *        *              *  *        *
-----
DATABASE              TDTEAM76              0
INDEX      TEAM76     TD76XA06              0
INDEX      TEAM76     TD76XB06              0
a TABLESPACE TDTEAM76 TD76TS06              0
VIEW      TEAM76     VWEMPPROJACT          0
VIEW      TEAM76     VWFORPLA              0
VIEW      TEAM76     VWSTAFAC2            0
***** END OF DB2 DATA *****
```

Figure 7-39 Alter the table space

Change LOG YES to LOG NO (Figure 7-40).

```

DB2 Admin ----- DB0B Redefine Table Space ----- Row 1 to 1 of 1
Command ==> Scroll ==> PAGE

Commands: CONTINUE ORIGINAL MAKEPBG MAKEPBR
Line commands: S - Split part R - Remove part O - Original data
               C - Clear data
CREATE TABLESPACE: TD76TS06 IN TDTEAM76

Numparts . . . . . 0 LOB . . . . . NO
Define . . . . . YES DSSIZE . . . . . LOG . . . . . no
Member Cluster . . NO SEGSIZE . . . . . 4 CCSID . . . . . EBCDIC
Buffer Pool . . . . BPO Close Rule . . NO Max Rows . . 255
Lock Size . . . . . ROW Lock Part . . . NO Lock Max . . 0
Max Partitions . . 0

S Part Pqty Sqty Page Free Compr R M T VCAT Stogroup
GBPCache
- -----> -----

0 -1 -1 0 5 NO N Y I DB0BD DSN8G100 CHANGED
***** END OF DB2 DATA *****

```

Figure 7-40 Redefine Table Space panel

You must either enter Continue on the Command line or click the Continue keyword at the top of the panel in order for the change to be recognized.

Note: If you press F3, your changes to the table space are lost.

Notice the Modification Pending notation next to the table space name (Figure 7-41).

```
DB2 Admin ----- DB0B Alter Related : TEAM76.TD76TB06_EMPPROJ Row 1 to 7 of 7
Command ==>                                           Scroll ==> PAGE
```

Commands: ADDINDEX ADDFK
Line commands: A - Alter object S - Show object R - Recreate view

S	Type	Qual	Name	Dep.	Views	Rel	Name	Note
*		*	*		*	*		*
-	-----	-----	-----	-----	-----	-----	-----	-----
	DATABASE		TDTEAM76		0			
	INDEX	TEAM76	TD76XA06		0			
	INDEX	TEAM76	TD76XB06		0			
*	TABLESPACE	TDTEAM76	TD76TS06		0			Modification pending
	VIEW	TEAM76	VWEMPPROJACT		0			
	VIEW	TEAM76	VWFORPLA		0			
	VIEW	TEAM76	VWSTAFAC2		0			
***** END OF DB2 DATA *****								

Figure 7-41 Alter Related with table space modification noted

To complete our change request, we need to specify an INCLUDE column in the TD76XA06 index. This index is related to the EMPPROJACT table and is one of the related objects listed on the Alter Related panel.

To change the index, enter an A (Alter Object) line command next to the index name (Figure 7-42).

```
DB2 Admin ----- DB0B Alter Related : TEAM76.TD76TB06_EMPPROJ Row 1 to 7 of 7
Command ==>                                           Scroll ==> PAGE
```

Commands: ADDINDEX ADDFK
Line commands: A - Alter object S - Show object R - Recreate view

S	Type	Qual	Name	Dep.	Views	Rel	Name	Note
*		*	*		*	*		*
-	-----	-----	-----	-----	-----	-----	-----	-----
	DATABASE		TDTEAM76		0			
a	INDEX	TEAM76	TD76XA06		0			
	INDEX	TEAM76	TD76XB06		0			
	TABLESPACE	TDTEAM76	TD76TS06		0			Modification pending
	VIEW	TEAM76	VWEMPPROJACT		0			
	VIEW	TEAM76	VWFORPLA		0			
	VIEW	TEAM76	VWSTAFAC2		0			
***** END OF DB2 DATA *****								

Figure 7-42 Alter Related

Include EMPTIME by issuing the I command next to the EMPTIME column name (Figure 7-43 and Figure 7-44 on page 204).

```
DB2 Admin ----- DB0B Redefine Index ----- Row 1 from 6
Command ==> Scroll ==> PAGE

Commands: CONTINUE ORIGINAL
Line commands: nnn A|D - Sequence & order R - Remove the column I - Include
A - Ascending D - Descending RA - Random U - Update expression/XML
pattern
B - Business Time without overlaps

CREATE INDEX TEAM76 . TD76XA06 >
ON TEAM76.TD76TB06_EMPPROJACT
Unique . . . . . YES Where Not Null . . . Cluster . . . . . NO
Buffer Pool . . . . BPO Close Rule . . . . . NO Copy Allowed . . NO
Piece Size . . . . . 2097152 Define . . . . . YES Defer . . . . .
Partitioned . . . . . Padded . . . . . Compress . . . . . NO

Select Column Name Col Type Length Scale N ColSeq Ord OldSeq Ord
* * * * *
-----
PROJNO CHAR 6 0 N 1 A 1 A
ACTNO SMALLINT 2 0 N 2 A 2 A
EMPNO CHAR 6 0 N 3 A 3 A
i EMPTIME DECIMAL 5 2 Y
EMSTDATE DATE 4 0 Y
EMENDATE DATE 4 0 Y
```

Figure 7-43 DB2 10 for z/OS INCLUDE Column Feature

```

DB2 Admin ----- DB0B Redefine Index ----- Row 1 from 6
Command ==> Scroll ==> PAGE

Commands: CONTINUE ORIGINAL
Line commands: nnn A|D - Sequence & order R - Remove the column I - Include
A - Ascending D - Descending RA - Random U - Update expression/XML
pattern
B - Business Time without overlaps

CREATE INDEX TEAM76 . TD76XA06 >
ON TEAM76.TD76TB06_EMPPROJACT
Unique . . . . . YES Where Not Null . . . Cluster . . . . . NO
Buffer Pool . . . . BPO Close Rule . . . . . NO Copy Allowed . . NO
Piece Size . . . . . 2097152 Define . . . . . YES Defer . . . . .
Partitioned . . . . . Padded . . . . . Compress . . . . . NO

Select Column Name Col Type Length Scale N ColSeq Ord OldSeq Ord
* * * * *
-----
PROJNO CHAR 6 0 N 1 A 1 A
ACTNO SMALLINT 2 0 N 2 A 2 A
EMPNO CHAR 6 0 N 3 A 3 A
* EMPTIME DECIMAL 5 2 Y 30 I
EMSTDATE DATE 4 0 Y
EMENDATE DATE 4 0 Y

```

Figure 7-44 DB2 10 for z/OS INCLUDE Column Result

Enter Continue on the Command line or click the **CONTINUE** keyword at the top of the panel. The Redefine Index - Space panel opens (Figure 7-45).

```

DB2 Admin ----- DB0B Redefine Index - Space ----- Row 1 to 1 of 1
Command ==> Scroll ==> PAGE

Commands: CONTINUE ORIGINAL
Line commands: 0 - Original data C - Clear data

CREATE INDEX TEAM76.TD76XA06
ON TEAM76.TD76TB06_EMPPROJACT
Sel Part Pqty Sqty FreePg %Free Erase ST VCAT Stogroup GBPCache
* * * * *
----->-----
0 -1 -1 0 10 NO I DB0BD DSN8G100 CHANGED
***** END OF DB2 DATA *****

```

Figure 7-45 Redefine Index - Space panel

Based on our change request, there are no additional changes to make on this panel. Enter Continue on the Command line or click the **CONTINUE** keyword at the top of the panel. The DDL for the index opens (Figure 7-46).

[illegible]

Figure 7-46 INDEX DDL

Press F3 to return to the Alter Related panel (Figure 7-47).

```
DB2 Admin ----- DB0B Alter Related : TEAM76.TD76TB06_EMPPROJ Row 1 to 7 of 7
Command ==> Scroll ==> PAGE

Commands: ADDINDEX   ADDFK
Line commands: A - Alter object  S - Show object  R - Recreate view
```

S	Type	Qual	Name	Dep.	Views	Rel	Name	Note
*		*	*		*	*		*
-	-----	-----	-----	-----	-----	-----	-----	-----
	DATABASE		TDTEAM76		0			
*	INDEX	TEAM76	TD76XA06		0			Modification pending
	INDEX	TEAM76	TD76XB06		0			
*	TABLESPACE	TDTEAM76	TD76TS06		0			Modification pending
	VIEW	TEAM76	VWEMPPROJACT		0			
	VIEW	TEAM76	VWFORPLA		0			
	VIEW	TEAM76	VWSTAFAC2		0			

```
***** END OF DB2 DATA *****
```

Figure 7-47 Alter Related

Press F3 one more time to return to the Alter Tables panel (ADB27CA) (Figure 7-48).

```
ADB27CA n ----- DB0B Alter Tables ----- Row 1 to 5 of 6
Command ==> Scroll ==> PAGE

Commands:
  ALTER - generate jobs   ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options

Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76  > (Table Schema)
Name . . . . > (Table Name. ? to look up)

Sel Qual      Object      Object      T DB Name  TS Name      RI RI  FK  Chg
*            *            *            * *          *      Rels Add Add Rqd Oper
*            *            *            * *          *      *  *  *  *  *
----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->----->
```

Figure 7-48 Alter Tables

As mentioned previously, the E under the Chg Rqd column must be satisfied or removed to complete the change.

At the top of the Alter Tables panel there are two options: A for Alter Object and an E to Edit the DDL. The A (Alter) line command displays the original definition of the object. If you enter an A next to the view VWEMPDPT1, the columns we previously added, phone and cellphone, would not be displayed. The user can add them at this time.

To edit the view DDL directly, enter an E (Edit DDL) line command next to the view name. In this case, the columns previously added, phone and cellphone, would be visible in the DDL. For this request, we use the E (Edit) option (Figure 7-49).

ADB27CA n ----- DB0B Alter Tables ----- Row 1 to 5 of 6
Command ===> Scroll ===> PAGE

Commands:
ALTER - generate jobs ADDFK - Add Foreign Key-affected tables
OPTIONS - Change alter options

Line commands:
A - Alter Object D - Delete S - Select Object REL - Alter related
FK - Add Foreign Key-affected tables E - Edit DDL
RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76 > (Table Schema)
Name > (Table Name. ? to look up)

	Object	Object				RI	RI	FK	Chg
Se1	Qual	Name	T	DB Name	TS Name	Rel	Add	Add	Rqd Oper
*	*	*	*	*	*	*	*	*	*
---	----->	----->	-----	-----	-----	---	---	---	-----
	TEAM76	TD76XA06	X			NA	NA		MODIFY
	TDTEAM76	TD76TS06	S	TDTEAM76	TD76TS06	NA	NA		MODIFY
*EL	TEAM76	TD76TB06_EMPPROJAC	T	TDTEAM76	TD76TS06	0	NA	NA	NONE
*	TEAM76	TD76TB01_DEPT	T	TDTEAM76	TD76TS01	0	NA	NA	MODIFY
e	TEAM76	VWEMPDPT1	V			NA	NA	E	MODIFY

Figure 7-49 Edit the view

The DDL view shows the previously added columns. Because our additional changes did not include the addition of any columns from the DEPT or the EMP PROJACT table, press F3 to return to the Alter Tables panel (ADB27CA). Notice that the E is removed from the Chg Rqd column (Figure 7-50).

```
ADB27CA n ----- DB0B Alter Tables ----- Row 1 to 5 of 6
Command ==>                                     Scroll ==> PAGE

Commands:
  ALTER - generate jobs  ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . TEAM76  >          (Table Schema)
Name . . . .          > (Table Name. ? to look up)

      Object  Object
Sel Qual  Name      T DB Name  TS Name      RI RI  FK  Chg
      *      *      * *      *      *  *  *  *  *
----->----->----->----->----->----->----->----->
      TEAM76  TD76XA06      X      NA  NA      MODIFY
      TDTEAM76  TD76TS06      S TDTEAM76  TD76TS06      NA  NA      MODIFY
*EL TEAM76  TD76TB06_EMP PROJACT T TDTEAM76  TD76TS06      0 NA  NA      NONE
*   TEAM76  TD76TB01_DEPT      T TDTEAM76  TD76TS01      0 NA  NA      MODIFY
*   TEAM76  VWEMPDP1T1      V      NA  NA      MODIFY
```

Figure 7-50 Updated Alter Tables panel

To “apply” the change, enter the ALTER command on the Command line or click the **APPLY** keyword (Figure 7-51).

```

ADBPALT ----- ALTER - Build Analyze and Apply Job -----
Option ==>

Specify the following:

Worklist information:
  Worklist name . . . . . ALTER      (also used as middle qualifier in
DSNs)
  Prefix for data sets . . . TEAM76

Data set information:
  PDS final qualifiers . . . JCL.CNTL
  Member name . . . . . ADBALTER
  Delete member name . . . ADBDELET (Optional job to delete work data
sets)

Options:
  Generate online . . . . . NO        (Yes/No)
  Generate one job . . . . . YES      (Yes/No)
  Member name or prefix . . APPLY
  As work statement list . . YES      (Yes/No)
  Unload method . . . . . U          (Unload, Parallel unload)

Optional processes:
  Run CHECK DATA . . . . . NO        (Yes/No)

```

Figure 7-51 Build Analyze and Apply Job panel

If the “Show this panel prior to each use” prompt on the ALTER Analysis Options panel is set to YES, the ALTER Analysis panel opens. You could modify any of the parameters or simply press Enter to move to the Build Analyze and Apply Job panel.

Complete the information on the Build Analyze and Apply Job panel.

The first section contains the eight character “Worklist name”. This name acts as the default name of the work statement list member and as the second level qualifier for generated data set names used by the product when making a change. The “Prefix for data sets” option represents the ID that is used as the prefix for underlying data set names.

The next section of the parameter panel pertains to the PDS information where two jobs, ADBALTER and ADBDELETE, are generated:

- ▶ ADBALTER consists of two parts, ANALYZE and APPLY, as described in “ANALYZE” on page 168 and “APPLY” on page 170. ADBALTER produces a detailed report of the differences between the DB2 Catalog definition and the changes entered at the ISPF panel, and it produces either an MVS set of JCL in a PDS or the change control cards in a work statement list member. The submission of the MVS JCL or the execution of the work statement list member actually makes the change.
- ▶ ADBDELET is an optional job that could be used to clean up old data set names after a change has successfully completed.

The Options section on the Build Analyze and Apply Job parameter panel is where the user can specify options to manage the generation of the Apply JOB. If the “Perform Analysis in Batch” entry on the ALTER Analysis Option panel is set to YES, then the “Generate Online” value on the ALTER Build Analyze and Apply job panel is protected or disabled.

“Generate one job” determines if one job or multiple jobs that make the change are generated. APPLY is the default prefix for any job JCL member generated in the PDS.

If the “As Work Statement List” option is set to YES, then control cards are generated and stored in a WSL library. Otherwise, an MVS JCL job stream is built.

The unload method supports IBM Unload and parallel unload for partitioned tables.

Note: At the time of the writing of this book, DB2 High Performance Unload was not supported.

At the bottom of the panel there is a list of ancillary utilities that may be executed after the change has completed. These utilities include CHECK DATA, COPY, REORG / BEBUILD, RUNSTATS, and REBIND (Figure 7-52).

```

----- ALTER - Build Apply Job -----
Option ==>

Specify the following:

Member name or prefix . . APPLY
As work statement list . . YES      (Yes/No)
Unload method . . . . . U          (Unload, Parallel unload, HPU)

Optional processes:
Run CHECK DATA . . . . . NO      (Yes/No)
Run COPY . . . . . N              (after: Reload/Alter/Both/None)
Run REORG/REBUILD . . . . . N      (Mandatory, All relevant, None)
Run RUNSTATS . . . . . N          (after: Reload/Alter/Both/None)
Run REBIND . . . . . NO           (Yes/No)

Utility control options:
Use templates . . . . . YES        (Yes/No)
Use utility options . . . NO       (Yes/No)

BP - Change batch job parameters
TU - Specify TEMPLATE usage
UO - Customize utility options

```

Figure 7-52 ALTER Build Analyze and Apply Job (continued)

Under the “Utility control options” heading is an option to use templates and specify some utility options. Templates provide the ability to specify data set names that meet the naming standards of the enterprise, thereby eliminating the need to make any modifications to either the generated jobs or to the underlying DB2 Administration Tool skeletons. Users may also elect to pick and choose various utility control card options if specifying YES for the “Use utility options” keyword.

After the information on the panel is complete, press Enter. The “Specify Work Statement List” panel opens.

Note: At the time of this writing, DB2 Administration Tool is working differently than in the past. The name of the work statement list library was retained from one session to another. In the current implementation, each time the Specify Work Statement List panel opens, the WSL library name consists of three parts. The first qualifier is the user ID. The second qualifier is the eight character worklist name specified on the ALTER Build Analyze and Apply Job panel. The third and last qualifier is the constant WSL. Anytime the worklist name is changed on the ALTER Build Analyze and Apply Job parameter panel, a new work statement list library could be generated because if the library does not exist, DB2 Administration Tool automatically generates the data set.

To reuse an existing work statement list library, overwrite the generated work statement list library name and member name if necessary.

The ADBALTER JCL job stream is generated. Submit this job to perform the ANALYZE and generate the APPLY job as an MVS JCL job stream in a PDS or as a member of the work statement list library.

If run in batch, the ANALYZE produces a report that can be viewed by SDSF (Example 7-1 and Example 7-2 on page 212).

Example 7-1 Output report

Compare tablespace source(TDTEAM76.TD76TS06) and target(TDTEAM76.TD76TS06)

(A) Tablespace changed from LOGGED to NOT LOGGED

(D) Field CREATOR changed from TEAM76 to DBA104

Tablespace TDTEAM76.TD76TS06 will be dropped

Tablespace will be recreated

Compare table source(Team76.TD76TB01_DEPT) and target(Team76.TD76TB01_DEPT)

Column LOCATION

(A) Type changed from CHAR(20) to CHAR(25)

(A) Column BUILDING appended

Table will be altered

Compare table source(Team76.TD76TB02_EMP) and target(Team76.TD76TB02_EMP)

Source and/or target tablespaces use table-controlled partitioning

Comparison results may not reflect actual object differences

(D) Column CELLPHONE added

Table Team76.TD76TB02_EMP is partitioned and will be dropped by dropping table

Example 7-2 Output report (continued)

Compare table source(Team76.TD76TB06_EMPPROJACT) and target(Team76.TD76TB06_EMPP
Tables have identical column lists
Table Team76.TD76TB06_EMPPROJACT will be dropped by dropping the tablespace
Table will be recreated

Compare index source(Team76.TD76XA01_DEPT) and target(Team76.TD76XA01_DEPT)
No changes to Index

Compare index source(Team76.TD76XA01_EPA) and target(Team76.TD76XA01_EPA)
(A)Column EMPTIME added to included columns
Index Team76.TD76XA01_EPA will be dropped by dropping the tablespace
Index will be recreated because the base table will be dropped and recreated

To make the physical change on the DB2 catalog, the APPLY job must be executed. If stored in a work statement list member, the job can be run online or in batch. If the job is stored as an MVS JCL job in a PDS, then the user must submit the job.



The MIG line command

The DB2 Administration Tool enables you to migrate (or copy) DB2 object definitions, the data for the objects, and the catalog statistics for the objects from one DB2 subsystem to other DB2 subsystems. You can migrate any combination of this set of information (object definitions, data, and statistics) for DB2 databases, table spaces, and tables, as well as their dependent objects.

In this chapter, we describe the process of using the Migrate (MIG) command supplied by the DB2 Administration Tool for z/OS V10 and prior releases of the product.

We show how to use the process to streamline the creation of schema and database environments while making automatic changes to the definitions and site specific standard changes.

This chapter contains the following topics:

- ▶ What the Migrate command is
- ▶ Migrate versus Compare
- ▶ Migrate high level process
- ▶ Masks
- ▶ Migration example

8.1 What the Migrate command is

Migrate is a command that allows you to move (or copy) DB2 Objects, and optionally their data and catalog statistics, from one schema to another. The level of scope of a migration can range from a database, to an index down to the catalog statistics of an object. All dependant objects, such as authorities, MQTs, and views, are migrated.

Migrate is typically used for:

- ▶ Creating a separate DB2 Subsystem
- ▶ Moving test systems (or objects) through the application life cycle
- ▶ Moving statistics from one subsystem to another to facilitate application testing

Migrate generates all the JCL and DDL required to copy the DB2 Object and includes any conversion required to achieve the migration.

The result of running a migrate is an identical set of DB2 objects, after any requested changes have been applied, on either the local or a remote DB2 subsystem.

8.2 Migrate versus Compare

DB2 Administration Tool contains two methods for managing change in a DB2 Subsystem. The first is Migrate, which we discuss in this chapter, and the second is Compare, which is discussed in Chapter 9, “DB2 Object Comparison Tool enhancements” on page 235.

The main difference between the two processes is that Compare primarily compares two sets of DB2 Objects (source and target) and generates all the necessary jobs required to upgrade the target objects to the same level as the source tables. Data in the target tables can be converted and preserved. The input into Compare can be from a variety of sources (DDL, catalog, and versions scopes).

Migrate takes its input from the DB2 catalog only, and can move data from the original DB2 Objects to the new DB2 Object and provide any conversion to both the data and the DDL.

Both Migrate and Compare make use of the DB2 Administration Tool Mask function (see 8.4, “Masks” on page 215) to allow for extensive customization of the schema definition, and both provide a data conversion process and supply all jobs that are required to produce the desired results. Both methods can use Work Statement Lists (see Chapter 10, “Work statement lists” on page 259) as the method of implementing changes, although Migrate customization is undertaken after the WSL is generated through the clone feature (see “Using WSL and masking” on page 229).

Use Migrate to copy, or move, new DB2 Objects in isolation, and use Compare to promote DB2 Objects where the target environment already exist and the new, or changed, object is part of a larger application.

8.3 Migrate high level process

There are three main steps in the Migrate process

1. Selecting Objects for migration

You can choose to migrate DDL, data, or catalog statistics or any combination of the three. When migrating statistics, SQL statements are generated that modify the statistics. The statements have the qualifier of the target catalog that is specified on the panel, and relate to the objects that have been selected for migration.

2. Generating migrate jobs

This step produces all the jobs that are required to perform the migration. It includes some or all of the following (depending upon your selections):

- Jobs for producing the DDL
- Unloading the data
- Performing any conversions
- Creating the new objects
- Reloading the data
- Any other utilities that you have requested

This step can be performed either online or in batch and the output can either be further jobs or a Work Statement List.

3. Running generated jobs

Execute the generated jobs to implement the database changes. All jobs contain instructions for restarting in the event of failure

These steps are expanded upon in 8.5, “Migration example” on page 222.

The following restrictions apply:

- ▶ Databases without table spaces are not migrated.
- ▶ Table spaces with DEFINE NO that are empty do not have JCL generated to unload or load the data. DDL migration is done.
- ▶ Migrate does not drop explicitly created LOB table spaces. You must create the table spaces if required.

8.4 Masks

Masking is a powerful feature of the DB2 Administration Tool that allows extensive changes to be made to Object definitions. These changes can range from naming standard changes to changes in the space attributes of individual objects. Additionally, there is the ability to invoke a user exit to run user written REXX programs, enabling you to add site specific overwrite rules to provide additional flexibility and customization.

Masks are used by both the Migrate and the Compare functions, and can be shared by the processes.

You are given the option of creating, and using, masks during the Migrate and Compare processes. Alternatively, you can define the masks in advance and select them for use during the process. All masks defined are saved for future use in the respective repository.

8.4.1 Creating a mask

Masks can be defined and stored in two ways (Figure 8-1):

- ▶ The data set option
Described in “Creating masks using a data set” on page 216
- ▶ The Enhanced Change Management
Described in “Creating a mask using Enhanced Change Management” on page 218

Creating masks using a data set

The first method is by using a data set, preferably a partitioned data set. To select this method, on the Specify Mask panel, enter a data set name, and, optionally, a member name (Figure 8-1).

```
ADB2GENM ----- Specify Mask -----  
Command ==>  
  
Mask Table Entry:  
  Owner . . . . > (? to look up)  
  Name . . . . > (? to look up)  
Data Set:  
  Mask DSN . . . MASK(TEST)  
Options:  
  Edit Mask . . . (Yes/No)
```

Figure 8-1 Specifying mask definitions in a data set

An edit session opens and you see a panel that shows the various options that are available for selection. These options are all mask fields, but they can be split into two categories:

- ▶ Field changes (Figure 8-2)
- ▶ Overwrite changes (Figure 8-3)

All options are identical regardless of the method used to store the mask definitions,

```
Fields (hierarchy):
COLNAME
NAME
  DBNAME, TSNAME, TBNAME, IXNAME, UDFNAME,
  UDTNAME, COLLNAME, PKGNAME, PGMNAME, PLNNAME
  DBRMNAME, STPNAME, SFNAME, TGNAME, GRPNAME,
  VCATNAME, GBPNAME, TCNAME, PMNAME, MKNAME
  BPNAME
    TSBPNAME, IXBPNAME
  SGNAME
    TSSGNAME, IXSGNAME
AUTHID
SQLID
SCHEMA
  TBSHEMA, IXSCHEMA, PMSHEMA, MKSCHEMA
OWNER
  DBOWNER, TOWNER, TOWNER, IXOWNER
GRANTID
  GRANTOR, GRANTEE
ROLE
  DBROLE, TSROLE, TBROLE, IXROLE
XMLSCHID
Examples:
OWNER:ABC*,DEF*
NAME:PRE*,NPRE*
XMLSCHID:P01,P02
```

Figure 8-2 Mask options fields

```
Overwrite Syntax:
Field:inmask,Overwrite_value
Fields:      Overwrite values:
COMPRESS    YES,NO,REXX exit
SEGSIZE      n (4-64 must be multiple of 4),REXX exit
DSSIZE       nG,REXX exit
PRIQTY       n,n%,REXX exit (table spaces and indexes)
  TSPRIQTY   n,n%,REXX exit (table spaces only)
  IXPRIQTY   n,n%,REXX exit (indexes only)
SECQTY       n,n%,REXX exit (table spaces and indexes)
  TSSECQTY   n,n%,REXX exit (table spaces only)
  IXSECQTY   n,n%,REXX exit (indexes only)
DEFER        YES,NO,REXX exit (indexes only)
DEFINE       YES,NO,REXX exit (table spaces and indexes)
  TSDEFINE   YES,NO,REXX exit (table spaces only)
  IXDEFINE   YES,NO,REXX exit (indexes only)
HASHSPC      nK,nM,nG,REXX exit
TBINLOBL     n,REXX exit (tables only)
DTINLOBL     n,REXX exit (distinct types only)
```

Figure 8-3 Mask overwrite fields

The field masks are used to change values within the DDL and DML that is generated by the Migrate (or Compare) process. Fields are hierarchical, so if NAME is selected, it would apply to all fields below NAME in Figure 8-2 on page 217, that is, DBNAME, TSNAME, and so on. If you just want to change the database name, then you should use DBNAME. It is therefore important to select the correct level of the masking field when defining your masking fields and to put the most restrictive fields first, as shown in Example 8-1. Masks lines are executed from top down and expected names may be changed before arriving at what you thought would change the field

Example 8-1 Ordering mask fields

```
NAME:TEST3*,TEST4*
DBNAME:TEST3DB,SYSTEMDB
```

Example 8-1 results in the database name becoming TEST4DB (from TEST3DB) because the NAME change is executed first, which changes the database name first, so the database name no longer matches and is changed to SYSTEMDB. If the order is changed, then the results would be as expected, as the database name is changed first and no longer matches the NAME field when that line is executed.

The mask overwrites are definitions that alter the value of non-name related attributes of the original definition, that is, attributes such as space, SEGSIZE, and so on. The new values can be based upon the existing values, be explicitly defined, or for some attributes, a user written REXX can be invoked to calculate a new value. The fields that can be changed through a REXX are COMPRESS, SEGSIZE, DEFER, DSSIZE, PRIQTY and SECQTY. For example, you could change the sizing of objects depending upon the destination, which could be based on the naming standards of the environment, which means that you could automate the sizing process during the Migration and Compare processing.

There have been new fields added to the support functionality being delivered in DB2 10 for z/OS, namely PMNAME, MKNAME, PMSHEMA, MKSCHEMA, and XMLSCHID for mask types, and HASHPC, TBINLOBL, and DTINLOBL for mask overwrites. A full list and definition of mask fields is shown in “Mask translation names” on page 504.

Creating a mask using Enhanced Change Management

The second method of creating a mask is by using a feature delivered with the introduction of Enhanced Change Management (ECM). This method uses DB2 tables to store the mask definition. You use a set of ISPF panels to enter data. To select this method, select the Mask Table Entry (Figure 8-4).

ADB2GENM ----- Specify Mask -----	
Command ==>	
Mask Table Entry:	
Owner	ADMR3 > (? to look up)
Name	MIGRATEMASK2 > (? to look up)
Data Set:	
Mask DSN . . .	
Options:	
Edit Mask . .	(Yes/No)

Figure 8-4 Specifying mask definitions in DB2

To use this option, Enhanced Change Management has to be enabled. To enable the ECM, you need to define the required DB2 Objects, found in member ADBCHANG in the SADBBSAMP library, and enter the creator of the tables in the CMOWN parameter in the ADBL CLIST found in the SADBCLST library. In addition, the CM - Change Management option is added to the main DB2 Administration Tool panel to allow management of the objects and to enter the ECM processes, which allows the creation of masks (and other DB2 Administration Tool features, such as Ignores) to be undertaken outside of the MIG process. For more details, see *DB2 for z/OS Administration Tools for Enhanced Change Management*, SG24-7441.

After entering the mask name and owner, an initial entry panel opens (Figure 8-5).

```

ADB2C22 n ----- CM - Insert Mask ----- 13:19
Command ==>

Owner . . . ADMR3   > (Optional, default is ADMR3, ? to lookup)
Name . . . MIGRATEMASK2   > (Required, ? to lookup)
Comment . . .                                     >

```

Figure 8-5 Entering the mask name

Here you can enter comments to describe the mask function, which should be a meaningful description, such as “Migrate from Test to Systems Test”, as masks are shareable and need only be defined once.

Tip: To allow easy reuse of masks, it is a best practice to use a naming standard that reflects both the source and the target in the name, as this allows easy identification of the correct mask.

Figure 8-6 shows the CM - Mask Lines panel, where you enter the mask line details, which is equivalent of adding the lines to the data set. Again, order is important, as the first line is executed first and so on.

```

ADB2C2L n ----- CM - Mask Lines ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> PAGE

Mask lines for mask "ADMR3"."MIGRATEMASK2"
Commands: CANCEL
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel   Sequence Type      From          To          Oper.  T
      * *              *              *              *    *
---  -----> -----> -----> -----> ----->
      1 ?              ?              ?

***** END OF DB2 DATA*****

```

Figure 8-6 Initial mask detail entry panel

You can enter individual lines of masking, repeat lines, move, and copy. The help panel for this panel contains all the details of the definitions that you can use along with details of panel specific commands. When you exit this panel, the definitions are saved unless you exit using the CANCEL command. Figure 8-7 shows how Example 8-1 on page 218 would be entered using this method; note that the order has changed.

```

ADB2C2L n ----- CM - Mask Lines ----- Row 1 from 2
Command ==> Scroll ==> PAGE

Mask lines for mask "ADMR3"."MIGRATEMASK2"
Commands: CANCEL
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel      Sequence Type      From      To      Oper.  T
      * *      *      *      *      *
----->----->-----
*          1 DBNAME    TEST3DB      SYSTEMDB      UPDATE
*          2 NAME      TEST3*      TEST4*      UPDATE
***** END OF DB2 DATA *****

```

Figure 8-7 Mask detail lines

In Figure 8-8, you make further changes or find information about the mask definition. The ML option takes you back to the detail lines for further editing, while the option E allows you to update the mask using the ISPF editor.

```

ADB2C21 n ----- CM - Masks ----- Row 1 to 1 of 1
Command ==> Scroll ==> PAGE

Line commands:
  U - Update  DEL - Delete  INS - Insert  ML - Mask lines  CH - Changes
  E - Edit   I - Details on mask

Sel      ID Owner      Name      Comment
      * *      *      *
-----
          1 ADMR3      MIGRATEMASK2      MIGRATE
***** END OF DB2 DATA *****

```

Figure 8-8 Mask lists

To edit the mask without using Migrate or Compare, you need to select CM option from the main DB2 Administration Tool. The panel shown in Figure 8-9 opens.

```
ADB2C min ----- Change Management (CM) ----- 14:05
Option ==>

1 - Manage changes
2 - Manage masks
3 - Manage ignores
4 - Manage versions
5 - Manage ID table
6 - Report changes

DB2 System: DB0B
DB2 SQL ID: ADMR3
CM Owner : ADB
```

Figure 8-9 Managing a mask using Change Management

Select Option 2 to open the Manage Masks panel (Figure 8-10).

```
ADB2C2 in ----- CM - Manage Masks ----- 14:26
Option ==>

1 - Display masks
2 - Create a mask

DB2 System: DB0B
DB2 SQL ID: ADMR3

Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . > Created by . . . . . >
Owner . . . . . ADMR3% > Altered by . . . . . >
Created within Mask ID . . . . .
Altered within
```

Figure 8-10 Selecting a mask

After completing the Owner field, a list of masks that match the selection criteria appears (Figure 8-8 on page 220). Here you can edit the masks.

Tip: You should use Enhanced Change Management to manage the objects related to making changes within DB2 Administration Tool, such as masks, Ignores and versions. ECM lets you use the Change Management system with DB2 managing the definitions, giving you one place that contains all objects required by the process.

Tip: Using many masks might affect performance. If a match is not found early in the process, the program must search through the list of translation masks until a match is found.

8.5 Migration example

The migration process flow is shown in Figure 8-11. Before starting a migration, you need to decide how you are going to generate the jobs, for example, use Work Statement Lists or generate in batch. In this example, we generate the migration online and use JCL.

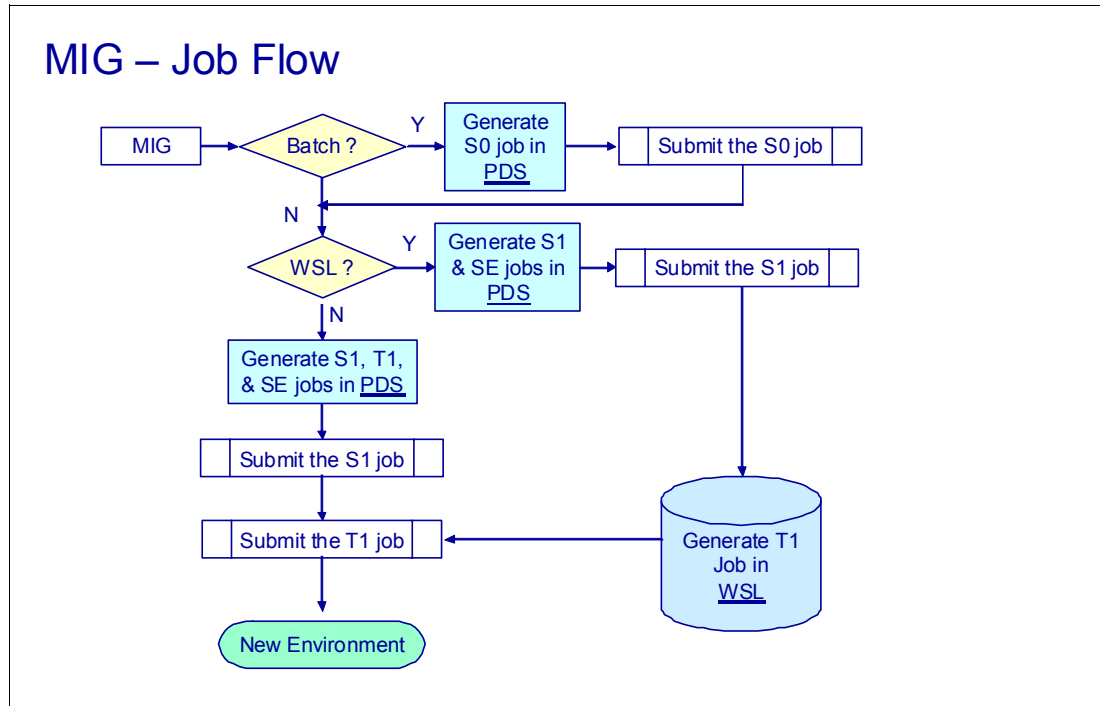


Figure 8-11 Migration process flow

8.5.1 Scenario

We have a database that is a “master” database that needs to be migrated to create a testing environment. The environment consists of one database (TDTEAM76), 14 table spaces, 14 tables, 16 indexes, and 24 views. The Schema is TEAM76. The new environment contains the same structure with a database name of ADMR3ADB and Schema of ADMR3. Additionally, the user needs a plan_table that is migrated from a different Schema (DB2R9)

This section shows you the steps for setting up the migration process to complete these activities.

Selecting the source objects

For this example, we are migrating at the database level and adding an additional object for the plan_table.

To select the database, query the catalog from the main DB2 Administration Tool panel using the database name. The Databases panel opens (Figure 8-12).

```
ADB21D in ----- DB0B Databases ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
DIS - Display database  STA - Start database  STO - Stop database  A - Auth
? - Show all line commands

Select Name      Owner      Storage  Buffer      Created      Index
      *        *        Group   Pool      DBID By      T E BPool   I
-----
      TDTEAM76 ADMR1      DSN8G100 BPO      408 ADMR1      E BPO      N
*****
***** END OF DB2 DATA *****
```

Figure 8-12 Database panel

On this panel, enter MIG as the line command (as there is only one object in the list, we could have used the Command line instead). A panel, shown in Figure 8-13, opens and contains all the objects selected for migration. If you want to exclude objects from this list, place a “-” (minus) character next to the object to be removed (this object will not be part of the migration process). If you remove one or all of the partitions of a partitioned table space, the complete table space is removed. If you have removed objects by accident, then the list can be reinstated by using the **refresh** command, which will reinstate the original list.

```
ADB28S in ----- DB0B Migrate Table Spaces ---- Row 1 to 10 of 19
Command ==>                                     Scroll ==> PAGE

Commands: MIG - Generate jobs  ADD - Add databases
          MIGRI - Add all RI related table spaces
Line commands:
D - Delete  T - Show tables  ADDRI - Add RI related table spaces
RIT - Show RI related tables

      Data      Table      No of      Table RI      VSAM
Select Base     Space     Part Tables LOB   Relations Added  KB Used
      *        *        *    * *    *    *    *
-----
      TDTEAM76 TD76TS01      0      1 NO      0 NA      720
      TDTEAM76 TD76TS02      1      1 NO      0 NA      720
      TDTEAM76 TD76TS02      2      1 NO      0 NA      720
      TDTEAM76 TD76TS02      3      1 NO      0 NA      720
      TDTEAM76 TD76TS02      4      1 NO      0 NA      720
      TDTEAM76 TD76TS02      5      1 NO      0 NA      720
      TDTEAM76 TD76TS03      0      1 NO      0 NA      720
      TDTEAM76 TD76TS04      0      1 NO      0 NA      720
      TDTEAM76 TD76TS05      0      1 NO      0 NA      720
      TDTEAM76 TD76TS06      0      1 NO      0 NA      720
```

Figure 8-13 Objects for migration

Now we add the plan_table. From the panel shown in Figure 8-13 on page 223, select the ADD databases. The Migrate Add Database panel opens, where you can add the databases to the migration list. You can enter a partial database name, as shown in Figure 8-14, and you are presented with a list to select from (Figure 8-15).

```
ADB28AD n ----- DB0B Migrate Add Database ----- 18:48
Option ==>

Enter the partial name of the database you want to add to the migrate
operation:

Partial database name . . : DSN%

Press Enter to search for the database.
```

Figure 8-14 Adding another object

Select the object that you require (in this case, DSN00015). The object is added to the list of objects and you are returned to the panel shown in Figure 8-13 on page 223, but with the extra object added.

```
ADB28ADD ----- DB0B Migrate Add Database -- Row 14 to 26 of 150
Command ==>                               Scroll ==> PAGE

Line commands:
S - Select

      Data
Select Base
      *
-----
s      DSN00014
      DSN00015
      DSN00016
      DSN00017
      DSN00018
      DSN00019
      DSN00020
      DSN00021
      DSN00022
      DSN00023
      DSN00024
      DSN00025
      DSN00026
```

Figure 8-15 Adding a database

Selecting the migration options

Enter MIG. The panel shown in Figure 8-16 opens, which contains all the information required to generate the jobs required to execute the migration.

```

ADB28M in ----- DB0B Migrate Parameters ----- 19:34
Option ==>

Please specify the following for DB2 Admin Migrate:      DB2 System: DB0B
                                                        DB2 SQL ID: ADMR3
Worklist name . . . . . : MIGTEST  (also used as middle qualifier in DSNs)

Data set information:
  PDS for jobs . . . . . : JCL.CNTL
  Prefix for datasets . . . : ADMR3

Target system parameters:
  DB2 subsystem id (SSID) . : DB0B      DB2 release . . . . :
  Target system node name . : DB0B      Submit job at local. : NO  (Yes/No)
  DB2 sample pgm load lib . :
  Use Cust. Table settings instead of the following Target libs: NO  (Yes/No)
    DB2 Admin APF library . :
    DB2 exit library . . . :
    DB2 load library . . . :
  New TS storage group . . :      > New IX storage group :      >
  New database . . . . . :      > New schema of objects:      >
  New grantor . . . . . :      >
  Catalog qualifier . . . . :      > (default SYSIBM , only applicable
                                     when scope contains catalog stats.)

Migrate options:
  Generate MIG jobs in batch . . . : NO      (Yes/No)
  Generate work stmt list . . . . : NO      (Yes/No)
  Use masking for batch migrate . . : YES     (Yes/No, N if stmt list is Y)
  Combine job steps . . . . . : YES     (Yes/No)
    Member prefix for combined jobs : M2     (default ADBMG )
  Scope of migrate:
    DDL . . . . . : YES     (Yes/No)
    Data . . . . . : YES     (Yes/No)
    Catalog statistics . . . . . : NO     (Yes/No)
  DROP on target before CREATE . . : NO     (Yes/No, No if scope DDL is NO)
  Create storage group . . . . . : NO     (Yes/No)
  Generate GRANT statements . . . . : YES     (Yes/No)
  Run SQLID . . . . . : ADMR3     (Blank, a SQLID, or <NONE>)
  Unload method . . . . . : U      (U - Unload)
  Parallel utilities . . . . . : NO     (Yes/No)

Optional steps after reload:
  Run CHECK DATA . . . . . : NO     (Yes/No)
  Run RUNSTATS . . . . . : NO     (Yes/No)
  Run IMAGE COPY . . . . . : NO     (Yes/No)
  Run REBIND . . . . . : NO     (Yes/No)

Utility control options:
  Generate template statements . . : NO     (Yes/No)
  Use customized utility options . : NO     (Yes/No)

BP - Change batch job parameters
UO - Customize utility options

```

Figure 8-16 Migration Parameters panel

One of the main parameters to point out is the option to use the customized libraries or to define your own libraries. This option is useful when you are running on another system that has differently named libraries, as it removes the need for any editing. By using the “Use Cust Table Settings.....”, you can toggle the use of the libraries on and off without erasing the library names. If one or more of the fields are empty, and you have selected to use the libraries, then DB2 Administration Tool will fall back to the customized values of the libraries.

We are using masks, which we define and are stored inside DB2 using the Enhanced Change Management facility. The masks make the changes as described in the 8.5.1, “Scenario” on page 222. The mask definition is shown in Figure 8-17.

Figure 8-17 Mask example

[illegible]

Figure 8-18 Generated combined Migrate jobs

The Migrate process generates between three and n jobs, depending upon your options and the number of jobs steps required to perform the Migrate. In this case, only three jobs were needed, as we selected to combine jobs.

If we had not selected that option, then we would have had the jobs generated that are shown in Figure 8-19.

[illegible]

Figure 8-19 Generated single migrate jobs

The job names are created according to the naming standards shown in Table 8-1.

Table 8-1 Possible job names

Job name	Generated when	Job function
xxxxxS0	In batch	Generates the jobs necessary to perform the migration. Only shown when Generate in Batch is selected.
xxxxxSE	Combined	Informational. Shows which data sets are required at the target site. Cleans up the job to remove data sets that are no longer required after completion.
xxxxxS1	Combined	source job. Generates DDL, unloads data, and performs any masking and conversion of data or DDL.
xxxxxT1	Combined	Target job. Creates objects, loads data, and runs any utilities.
SST1RE	Single	Retrieves mask (if stored in DB2) and generates DDL.
SST2UL1	Single	Unloads data.
SST3CH	Single	Retrieves the mask and performs any required conversion.
SST4XF	Single	Informational. Shows which data sets may require transferring to the target site.
SST5DE	Single	Cleans up unwanted data sets on the source system after completion.
TST1CR	Single	Creates a new environment on the target.
TST2RL1	Single	Reloads data.
TST3CK	Single	Run Check Data.

Job name	Generated when	Job function
TST4RS	Single	Runs Runstats.
TST5IC	Single	Runs Image Copy.
TST6RB	Single	Runs Rebind.
TST7DE	Single	Cleans up unwanted data sets on the target system after completion.

xxxxx is the prefix entered in the Migration Parameters panel. Single jobs prefixed with SS are used with the source subsystem, while jobs prefixed with TS are used for the target subsystem. The order of submission is shown by the middle numeric of the single jobs and the trailing number of the combined jobs. When multiple jobs are required for a process, a numeric suffix is added to the job name and incremented as required.

Not all jobs are generated; it depends upon the options selected. In our case, we did not choose to have any utilities generated.

We run the jobs we created and perform the migration. As part of the output from the Migrate process, a report showing the result of the create statements is generated (Example 8-2). This example has been shortened for space reasons.

Example 8-2 Create Report from Migrate example

***** TOP OF DATA *****

2010-10-13 - DB2 Administration Tool - Summary Report for DB2BATCH

Time	Ret Code	Action	Object
=====	=====	=====	=====
19:26.07	0	CREATE	TABLESPACE TD76TS01
19:26.07	0	CREATE	TABLE TD76TB01_DEPT
19:26.07	0	CREATE	UNIQUE INDEX TD76XA01 ON ADMR3.TD76TB01_DEPT
19:26.07	0	CREATE	INDEX TD76XB01 ON ADMR3.TD76TB01_DEPT
19:26.08	0	CREATE	INDEX TD76XC01 ON ADMR3.TD76TB01_DEPT
19:26.08	0	CREATE	TABLESPACE TD76TS02
19:26.08	0	CREATE	TABLE TD76TB02_EMP
19:26.09	0	CREATE	UNIQUE INDEX TD76XA02 ON ADMR3.TD76TB02_EMP
19:26.09	0	CREATE	INDEX TD76XB02 ON ADMR3.TD76TB02_EMP
19:26.09	0	CREATE	TABLESPACE TD76TS03
19:26.09	0	CREATE	TABLE TD76TB03_ACT
19:26.09	0	CREATE	UNIQUE INDEX TD76XA03 ON ADMR3.TD76TB03_ACT
19:26.09	0	CREATE	UNIQUE INDEX TD76XB03 ON ADMR3.TD76TB03_ACT
.....			
19:26.10	0	CREATE	TABLESPACE TD76TS06
19:26.10	0	CREATE	TABLE TD76TB06_EMPPROJACT
19:26.10	0	CREATE	INDEX TD76XB06 ON ADMR3.TD76TB06_EMPPROJACT
19:26.10	0	CREATE	UNIQUE INDEX TD76XA06 ON ADMR3.TD76TB06_EMPPROJACT
19:26.10	0	CREATE	TABLESPACE TD76TS07
19:26.10	0	CREATE	TABLE TD76TB07_EACT
19:26.10	0	CREATE	TABLESPACE TD76TS08
19:26.10	0	CREATE	TABLE TD76TB08_EDEPT
.....			
19:26.12	0	CREATE	TABLESPACE TD76TS14
19:26.12	0	CREATE	TABLE TD76TB14_DEPT
19:26.12	0	CREATE	UNIQUE INDEX TD76XA14 ON ADMR3.TD76TB14_DEPT

```

19:26.12      0 CREATE   INDEX TD76XB14 ON ADMR3.TD76TB14_DEPT
19:26.12      0 CREATE   INDEX TD76XC14 ON ADMR3.TD76TB14_DEPT
19:26.12      0 CREATE   TABLE PLAN_TABLE
19:26.12      0 CREATE   VIEW TD76DEPMG1
19:26.12      0 CREATE   VIEW
19:26.12      0 CREATE   VIEW TD76EMPDP1
19:26.12      0 CREATE   VIEW TD76EMPDP2
.....
19:26.12      0 CREATE   VIEW TD76ASTRDE2
19:26.12      0 CREATE   VIEW VWASTRDE1
19:26.12      0 CREATE   VIEW VWASTRDE2
19:26.12      0 CREATE   VIEW VWFORPLA
19:26.12      0 CREATE   VIEW VWPSTRDE1
19:26.12      0 CREATE   VIEW VWPSTRDE2
19:26.12
19:26.12      End of Summary Report
***** BOTTOM OF DATA *****

```

Another point to make is that we had a mask override for the database for the plan_table from DSN00015 to ADMR3ADB. The database for the new table is *not* changed to ADMR3ADB, as shown in Figure 8-20, because the database is implicitly created and therefore not changed, as it is not in the generated DDL.

```
ADB21T in ----- DBOB Tables, Views, and Aliases ----- Row 12 from 40
Command ==>
Scroll ==> PAGE

Commands: GRANT MIG ALL

Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands
```

Sel	Name	Schema	T	DB Name	TS Name	Cols	Rows	ChksC
*	*	*	*	*	*	*	*	**
-----	-----	-----	-	-----	-----	-----	-----	-----
	TD76TB11_EPROJACT	ADMR3	T	ADMR3ADB	TD76TS11	7	-1	0
	TD76TB12_EEMP	ADMR3	T	ADMR3ADB	TD76TS12	16	-1	0
	TD76TB13_PARTS	ADMR3	T	ADMR3ADB	TD76TS13	4	-1	0
	TD76TB14_DEPT	ADMR3	T	ADMR3ADB	TD76TS14	5	-1	0
	PLAN_TABLE	ADMR3	T	DSN00148	PLANRTAB	64	-1	0
	TD76DEPMG1	ADMR3	V	ADMR3ADB	TD76TS14	7	-1	0
	TD76DEPT	ADMR3	V	ADMR3ADB	TD76TS14	4	-1	0
	TD76EMPDP1	ADMR3	V	ADMR3ADB	TD76TS14	7	-1	0
	TD76HDEPT	ADMR3	V	ADMR3ADB	TD76TS14	5	-1	0
	VWACT	ADMR3	V	ADMR3ADB	TD76TS03	3	-1	0
	VWDEPMG1	ADMR3	V	ADMR3ADB	TD76TS01	7	-1	0

Figure 8-20 Implicit database names

The migration is now completed and the SE job can now be run to clean up the data sets that are no longer required.

Using WSL and masking

As discussed previously, you have the option to generate batch jobs or Work Statement Lists (WSL) to carry out the Migrate process. We have seen how the batch jobs are generated and how masks are used. In this section, we briefly look at how to use WSL and how to use masks with them. We will not discuss WSL in detail, as that discussion is covered in Chapter 10, “Work statement lists” on page 259.

When we use WSL, a batch job (S1) is generated to run on the source system (the cleanup job SE is also generated but not discussed here). This job is similar to the one generated by the batch process except that it has two extra steps: The first step combines all the LOAD statements with the LOAD templates, and the second step creates the WSL and populates it with all the necessary statements to successfully complete the migration (DDL, DML, LOADS, utilities, and so on).

To view the WSL, either select W from the main menu, or enter WSL on any Command line. The window shown in Figure 8-21 opens.

```
ADB2W min ----- DB0B Manage Work Statement Lists ----- 13:53
Option ==>

      1 - Show work statement list library          DB2 System: DB0B
      2 - Show work statement list                  DB2 SQL ID: ADMR3

Work stmt list dsn ==> WSL.CNTL
Work stmt list name ==>
```

Figure 8-21 Initial Work Statement List panel

Selecting option 1 gives you a list of WSLs that are in your WSL data set (Figure 8-22). Alternatively, you can enter a member name and use option 2 to jump straight to the WSL content (selecting option S performs the same action (Figure 8-22)).

```
ADB2W1 in ----- Work Statement List Library: WSL.CNTL -- Row 1 to 1 of 1
Command ==>                                     Scroll ==> PAGE

Line commands:
S - Show  R - Run in batch D - Delete C - Copy  A - Append Q - Clone
I - Interpret V - Validate E - Edit  O - Run online

Sel Name      Created      Changed      ID
*             *             *             *
-----
MIGEXAMP 2010/10/13 2010/10/13 20:19 ADMR3
***** END OF DB2 DATA*****
```

Figure 8-22 WSL library members

The mask has *not* been applied to the WSL and contains the necessary commands to recreate the source environment. Masks cannot be selected in the migration panel if WSL is the selected method. So, you have to clone the WSL using option Q. The panel shown in Figure 8-23 opens, where you can specify the clone name (a new PDS member) and the mask that you want to use.

```

ADB2W1Q n ----- Clone Work Statement List -----
Command ==>                                         Scroll ==> PAGE

Input work stmt list information:                    DB2 System: DB0B
Work stmt list . . . : MIGEXAMP                     DB2 SQL ID: ADMR3
from library . . . . : WSL.CNTL

                                                    More:      +

Output work stmt list information:
Library (PDS name) . . : WSL.CNTL
Work stmt list . . . : MIGEXAM1 (will be new PDS member)

Execution mode . . . . : TSO      (BATCH or TSO)
PDS for jobs . . . . :
PDS member . . . . :
Unit type . . . . . : SYSDA

Use Masking. . . . . : YES      (Yes/No)
Apply masking to data set names. . : NO      (Yes/No=default)

Use local DB2 catalog information to replace: (Yes/No)
Authorizations . . . . . : NO
Partitioning attributes . . . . . : NO
Table space and index attributes . : NO

```

Figure 8-23 WSL cloning using masking

You specify the new WSL (in this case, MIGEXAM1) and that you want to use masking. The mask panels shown in Figure 8-4 on page 218 open. When you have selected your mask, press Enter and the WSL is cloned and masking is applied. Refresh the WSL list and the new WSL appears (Figure 8-24).

```

ADB2W1 in ----- Work Statement List Library: WSL.CNTL -- Row 1 to 2 of 2
Command ==>                                         Scroll ==> PAGE

Line commands:
S - Show  R - Run in batch  D - Delete  C - Copy  A - Append  Q - Clone
I - Interpret  V - Validate  E - Edit  O - Run online

Sel Name      Created      Changed      ID
*          *          *          *
-----
MIGEXAMP 2010/10/13 2010/10/13 20:19 ADMR3
MIGEXAM1 2010/10/14 2010/10/14 14:10 ADMR3
***** END OF DB2 DATA *****

```

Figure 8-24 WSL Library after cloning

To see what is in the WSL, either use the Show command, or to get a summary of objects, utilities, and commands use the Interpret command. After selecting the Interpret command, a panel opens that allows you to select only those features you are interested in seeing (Figure 8-25).

```

ADB2W10 n ----- Interpret Work Statement List Options -----

Specify S to select Work Statement List statement types:

SQL:                                DB2 Utilities:                DB2 Commands:                More:      +
  S DDL                                S Load/Unload                S Plan/packages
    ALTER                                LOAD                            BIND
    CREATE                            UNLOAD/REORG UNLOAD            REBIND
    DROP                                S Backup/Recovery            FREE
    COMMENT ON                        COPY                            S Other
    LABEL ON                          COPYTOCOPY                    RUN
    SET                              MERGECOPY                    START/STOP
  S DCL                                MODIFY                        Other
    GRANT                            QUIESCE                        Admin:
    REVOKE                            REBUILD                      S Data set
  S DML                                RECOVER                      ALLOC
    DELETE                            REPORT                      TSODELETE
    INSERT                            S Other                      LISTDEF
    UPDATE                            CHECK                        TEMPLATE
    Other                            DIAGNOSE                      ADBSYSIN
    COMMIT                            REORG                        S Other
    Comments                        REPAIR                      ADBPAUSE
  S Other                            RUNSTATS                    UTILFROM
                                STOSPACE                    REXX Execs

```

Figure 8-25 Interpreting the WSL

When you have selected the features in which you are interested, press Enter and the WSL details are presented (Figure 8-26). This is just a sample of the output due to the number of objects being migrated.

ADB2W1I n ----- Interpret Work Statement List: MIGEXAM1 Row 1 to 16 of 165						
Command ==>				Scroll ==> PAGE		
Line commands: S - Show object V - View statement						
Sel	Seq	Action	Object Type	Qual	Name	Note
*	*		*	*	*	*
----->-----						
	71	SET	SQLID	ADMR3		
	83	CREATE	DATABASE		ADMR3ADB	
	92	CREATE	TABLESPACE	ADMR3ADB	TD76TS01	
	100	CREATE	TABLE	ADMR3	TD76TB01_DEPT	
	109	CREATE	INDEX	ADMR3	TD76XA01	Unique
	118	CREATE	INDEX	ADMR3	TD76XB01	
	127	CREATE	INDEX	ADMR3	TD76XC01	
	136	CREATE	TABLESPACE	ADMR3ADB	TD76TS02	
	144	CREATE	TABLE	ADMR3	TD76TB02_EMP	
	153	CREATE	INDEX	ADMR3	TD76XA02	Unique
	162	CREATE	INDEX	ADMR3	TD76XB02	
	171	CREATE	TABLESPACE	ADMR3ADB	TD76TS03	
	179	CREATE	TABLE	ADMR3	TD76TB03_ACT	
	188	CREATE	INDEX	ADMR3	TD76XA03	Unique
					
	430	CREATE	TABLESPACE	ADMR3ADB	TD76TS14	
	438	CREATE	TABLE	ADMR3	TD76TB14_DEPT	
	447	CREATE	INDEX	ADMR3	TD76XA14	Unique
	456	CREATE	INDEX	ADMR3	TD76XB14	
	465	CREATE	INDEX	ADMR3	TD76XC14	
	480	CREATE	TABLE	ADMR3	PLAN_TABLE	
	488	SET	SCHEMA		ADMR3	
	491	SET	PATH		"SYSIBM", "SYSFUN"	
	493	CREATE	VIEW	ADMR3	TD76DEPMG1	
	502	CREATE	VIEW	ADMR3	TD76DEPT	
					
	713	TEMPLATE	DATASET	SYSREC	ADMR3.MIGEXAMP.ULD	
	714	TEMPLATE	DATASET	SYSERR	ADMR3.MIGEXAMP.TES	
	715	TEMPLATE	DATASET	SYSUT1	ADMR3.MIGEXAMP.TES	
	716	TEMPLATE	DATASET	SORTOUT	ADMR3.MIGEXAMP.TES	
	717	TEMPLATE	DATASET	SYSMAP	ADMR3.MIGEXAMP.TES	
	718	LOAD	TABLE	ADMR3	TD76TB01_DEPT	
	721	TEMPLATE	DATASET	SYSREC	ADMR3.MIGEXAMP.ULD	
	722	TEMPLATE	DATASET	SYSERR	ADMR3.MIGEXAMP.TES	
	723	TEMPLATE	DATASET	SYSUT1	ADMR3.MIGEXAMP.TES	
	724	TEMPLATE	DATASET	SORTOUT	ADMR3.MIGEXAMP.TES	
	725	TEMPLATE	DATASET	SYSMAP	ADMR3.MIGEXAMP.TES	
	726	LOAD	TABLE	ADMR3	TD76TB02_EMP	
	729	TEMPLATE	DATASET	SYSREC	ADMR3.MIGEXAMP.ULD	
	730	TEMPLATE	DATASET	SYSERR	ADMR3.MIGEXAMP.TES	
	731	TEMPLATE	DATASET	SYSUT1	ADMR3.MIGEXAMP.TES	
	732	TEMPLATE	DATASET	SORTOUT	ADMR3.MIGEXAMP.TES	
	733	TEMPLATE	DATASET	SYSMAP	ADMR3.MIGEXAMP.TES	
					

Figure 8-26 Sample WSL Interpret output

This method of migrating would be suitable when a master environment, which changes infrequently, is cloned to many environments. By using the WSL, this environment could be cloned and masked to match the required environment. This action would alleviate the need for the source generate to be executed, assuming that the data unloaded is still valid. If not, a data only migration could be run. The WSL would only need to be updated when the master environment has changed.



DB2 Object Comparison Tool enhancements

This chapter describes some of the newer features of the DB2 Object Comparison Tool.

The chapter contains the following topics:

- ▶ Overview of DB2 Object Comparison Tool
 - Defining the source or target
 - Overview of Version Scopes
 - Discussion of masks
 - Discussion of Ignore
- ▶ COMPARE using Version Scope
 - Example using Version Scopes to do a compare of indexes
- ▶ COMPARE: Delta change file

9.1 DB2 Object Comparison Tool

DB2 Object Comparison Tool is an add on product to the DB2 Administration Tool. Its purpose is to compare the physical structure of two sets of DB2 objects. The results of this comparison consists of a set of reports that highlight the differences. In addition, compare provides an option to generate an APPLY job which, when executed, is used to synchronize the two environments, making them look the same from a physical structure perspective.

MIG versus COMPARE: Some users might confuse the difference between DB2 Administration Tool's MIG feature and DB2 Object Comparison Tools capabilities:

- ▶ MIG is used to create a new environment in the same DB2 subsystem or in another DB2 subsystem. If the scope of the migration includes DDL and data, upon completion of the MIG process the target is identical to the source in all respects.
- ▶ The APPLY job of Compare, conversely, makes the physical structure of the two environments synonymous. The target *will* contain its original data. The structures are the same, but the data values are different.

The compare process consists of five steps:

1. Define the source, which is a required step.
2. Define the target, which is a required step.
3. Define a mask, which is an optional step.
4. Define an ignore, which is an optional step.
5. Generate and submit the compare job.
 - a. Generate a set of compare reports.
 - b. Optionally, build the APPLY job.

9.1.1 Defining the source or target

Compare uses the terms source and target, to differentiate the two sets of objects to be compared. New users are sometimes challenged when trying to determine which set of objects should be the source and which should be the target. The answers to the following questions may help:

- ▶ Which set of objects have the most up to date definition?
- ▶ Which set of objects contains all of the changes that are needed?
- ▶ Which set of objects is "correct"?

The set of objects satisfying these questions is the source, making the other set of objects the target.

After the source and target have been identified, the next step is to identify the source or origin of the DDL definition for both the source and the target. Where does the DDL definition reside? Previous releases of the DB2 Object Comparison Tool had three options for the origin of the DDL:

- ▶ DDL stored in a sequential file or a PDS member
- ▶ The DB2 catalog
- ▶ A Compare version file

A new origin, the Version Scope, has been added to the DB2 Object Comparison Tool (Figure 9-1).

Note: If the DDL is stored in a sequential file, a PDS, in the DB2 catalog, or in a version scope, DB2 Object Comparison Tool extracts the DDL and stores it in a version file. There is one version file for the source and one for the target. Version files are in a proprietary format and can be used as a source for either the source or the target of a future Compare. When used as input, the extraction of the DDL step is skipped, resulting in a performance boost. Regardless of the origin of the DDL, the actual Compare is done between the two version files.

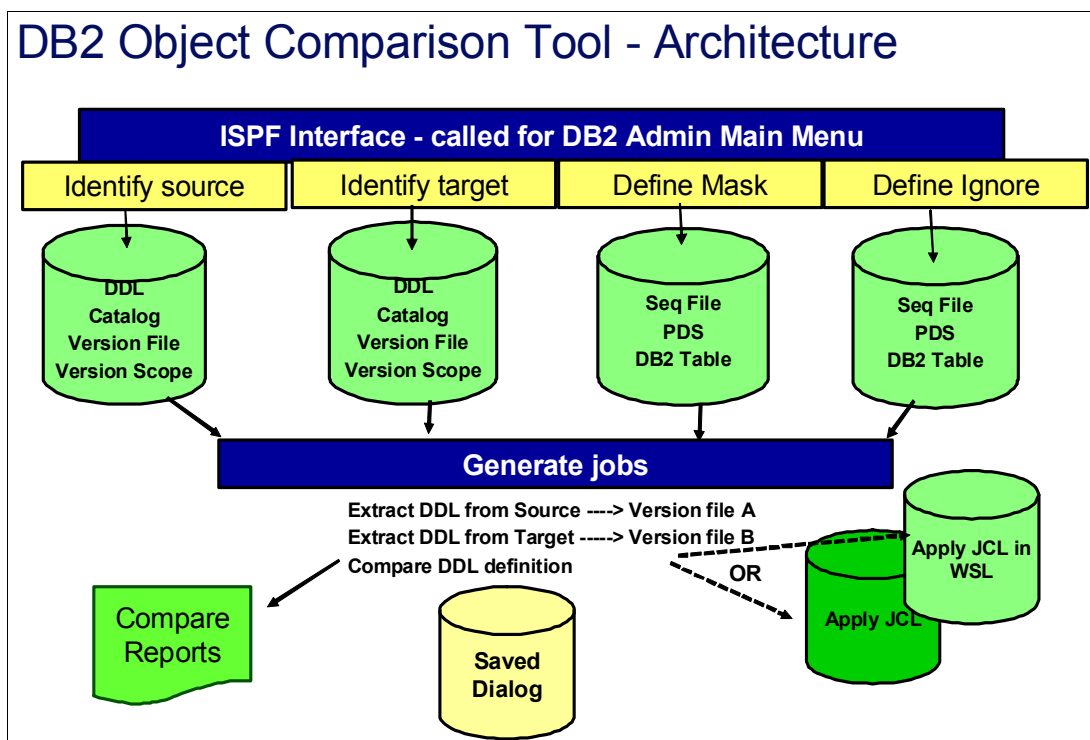


Figure 9-1 DB2 Object Comparison Tool architecture

9.1.2 Version Scope

Version scopes were introduced as a component of the DB2 Administration Tool V7.2 Enhanced Change Management facility.

Note: Enhanced Change Management requires the licensing of both the DB2 Administration Tool and DB2 Object Comparison Tool.

Version scopes are named entities, defined by the user, consisting of a list of DB2 objects. The definition of the version scope is stored in the DB2 Administration Tool's Enhanced Change Management repository. Version scopes are required to generate a base version. Their usage has been expanded and they can now be used as an origin for the definition of the source and target DDL of a compare.

Note: A base version is a snapshot of the object definitions at a point in time. They are used as input to the Enhanced Change Management PROMOTE process, which is actually a COMPARE type of change, used to propagate changes from one environment to another. During the PROMOTE, two base versions are compared (before and after) and any changes are placed in a Delta Version File. This file is IMPORTED into a DB2 target subsystem as an ANALYZED change ready for execution.

A multi-step process is used to define version scope using the DB2 Administration Tool change management panels:

1. Define the version scope:
 - a. Enter CM on the main DB2 Administration Menu or enter CMM as a primary command on any DB2 Administration Tool panel.
 - b. Select option 4 Manage Versions from the Change Management menu.
 - c. To create a new version scope, a new definition must be inserted into the change management repository. To perform this task, specify 3 Insert a version scope on the Option line of the Manage Versions panel. You will then be prompted to provide an owner, the name of the version scope, and a comment. Upon pressing Enter, an `INSERT stmt executed` message is displayed at the top of the panel under the Command line.
 - d. If the version scope exists and you want to update the object list, display the existing version scopes by entering a 2 on the Option line of the Manage Versions panel.
2. Populate the version scope:
 - a. Issue the SO line command (Version Scope Objects) next to the name of the version scope.
 - b. Insert the objects to be included. Press F1 to display a list of object types. Wildcarding is supported.
3. Optionally, use the GV command to generate a base version.

Version Scope and DB2 Object Comparison Tool

DB2 Object Comparison Tool V7.2 added the Version Scope as a potential source, or origin, for the DDL definition of both the target and source.

When used by Compare, the Version Scope does not have to be generated. In other words, Compare does not require a base version. Compare only needs the list of objects in the definition of the version scope.

DB2 Object Comparison Tool supports compares starting at the database, table space, and table levels. Using a version scope as the origin makes it possible to extend the capabilities of Compare. For example, a version scope, containing only a list of indexes, could be used to just compare the indexes defined on a table or data base.

9.1.3 Mask

Using a mask is optional when doing a compare. Its purpose is to align the owner and names of the objects in the source to those in the target, thereby ensuring that the right objects are being compared to each other. For example, if the source table is DEV.EMP and the target table is PROD.EMP, mask OWNER:DEV,PROD would indicate to Compare that the two tables should be compared even though they have different owners.

If the owners and names of the source and target are the same, then a mask is not needed.

Masks can be stored in a table in the change management repository, in a sequential file, or as a member of a PDS. When masks are stored in a table, they are easy to find and reuse, but they cannot be cloned. When stored in a sequential file or PDS, they can be easily cloned, but managing their location could be challenging if there are no naming standards in place.

9.1.4 Ignore

The purported goal of a comparison is to make the target look like the source. However, there are times when there are certain things that should not be changed. For example, when comparing development definitions to production, the development environment is often much smaller. Thus, you would not want Compare to make the PQTY and SQTY definitions in production match those in development. By ignoring PQTY and SQTY, the size of the production objects can be restored to their original definition.

Like masks, Ignore specifications can be stored in a table in the change management repository or in a file.

9.2 Compare using Version Scope

In the following scenario, we use Version Scopes as the origin for the DDL of both the source and target to demonstrate how to compare just the indexes between two environments. See 9.1.2, “Version Scope” on page 237 for more details about version scopes. When doing a comparison, the version scope definitions may already exist and can be used. However, in this example we are going to go through the process of creating a version scope for the source and another for the target.

To define a version scope, we must first issue the CM command on the Command line of the DB2 Administration Tool main menu or enter CMM as a primary command on any DB2 Administration Tool panel (Figure 9-2).

```
DB2 Admin ----- DB2 Administration Menu 10.1.0 ----- 19:41
Option ==> CM

    1 - DB2 system catalog                DB2 System: DBOB
    2 - Execute SQL statements            DB2 SQL ID: ADMR1
    3 - DB2 performance queries           Userid   : ADMR1
    4 - Change current SQL ID              DB2 Schema: ADMR1
    5 - Utility generation using LISTDEfs and TEMPLATES DB2 Rel   : 1015
    P - Change DB2 Admin parameters
    DD - Distributed DB2 systems
    E - Explain
    Z - DB2 system administration
    SM - Space management functions
    W - Manage work statement lists
    X - Exit DB2 Admin

    CM - Change management

                                         More:      +

Interface to other DB2 products and offerings:

    CP - DB2 Object Comparison Tool
```

Figure 9-2 DB2 Administration Tool main menu

On the Change Management panel, enter 4 - Manage versions on the Option line (Figure 9-3).

```
DB2 Admin ----- Change Management (CM) ----- 19:43
Option ==> 4

      1 - Manage changes
      2 - Manage masks
      3 - Manage ignores
      4 - Manage versions
      5 - Manage ID table
      6 - Report changes

DB2 System: DBOB
DB2 SQL ID: ADMR1
CM Owner  : ADB
```

Figure 9-3 Change Management panel

To create a new version scope, an initial definition must be inserted into the change management repository. To perform this task, enter 3 on the Option line of the Manage Versions panel (Figure 9-4).

```
DB2 Admin ----- CM - Manage Versions ----- 19:44
Option ==> 3

      1 - Display versions
      2 - Display version scopes
      3 - Insert a version scope
      4 - Import a version file

DB2 System: DBOB
DB2 SQL ID: ADMR1

Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . > Created by . . . . . >
Owner . . . . . > Altered by . . . . . >
Created within Version ID . . . . .
Altered within
```

Figure 9-4 Manage Versions panel

On the Manage Versions panel, specify an owner, the name of the version scope, and a comment (Figure 9-5).

Note: Because version scopes can be reused, providing a good name and a meaningful comment can be useful in the future.

```
DB2 Admin ----- CM - Insert Version Scope ----- 19:47
Command ==>

Owner . . . ADMR1 > (Optional, default is ADMR1, ? to lookup)
Name . . . SSEMMDBB INDEX > (Required, ? to lookup)
Comment . . INDEXES ONLY FOR SSEMMDBB >
```

Figure 9-5 Insert Version Scope skeleton

After the information has been entered, press Enter and an INSERT Statement Executed message will be displayed under the Command line at the top of the Insert Version Scope panel (Figure 9-6).

```
DB2 Admin ----- CM - Insert Version Scope ----- 19:49
Command ==>
INSERT stmt executed

Owner . . . ADMR1    > (Optional, default is ADMR1, ? to lookup)
Name . . . SSEMMDBB INDEX    > (Required, ? to lookup)
Comment . . INDEXES ONLY FOR SSEMMDBB    >
```

Figure 9-6 Insert Version Scope completed

At this point, the version scope is empty. Press F3 to return to the CM Manage Version panel.

The next step is to populate the version scope with a set of objects. Enter a 2 on the Option line of the CM Manager Versions panel to display version scopes (Figure 9-7).

```
DB2 Admin ----- CM - Manage Versions ----- 19:49
Option ==> 2

1 - Display versions                                DB2 System: DBOB
2 - Display version scopes                        DB2 SQL ID: ADMR1
3 - Insert a version scope
4 - Import a version file

Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . > Created by . . . >
Owner . . . . . > Altered by . . . >
Created within                                     Version ID . .
Altered within
```

Figure 9-7 CM Manage Versions

To add object(s) to a version scope, enter the SO Version scope objects line command next to the name of the version scope (Figure 9-8).

```
DB2 Admin ----- CM - Version Scopes ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> PAGE

Line commands:
VE - Versions SO - Version scope objects GV - Generate new base version
INS - Insert U - Update DEL - Delete I - Details on version scope

Sel          ID Owner    Name                      Comment
   *         * *        *                      *
----->
S0          3 ADMR1    SSEMMDBB INDEX          INDEXES ONLY FORSSEMMD
***** END OF DB2 DATA *****
```

Figure 9-8 CM Version Scope objects

```
DB2 Admin ----- CM - Version Scope Objects ----- Row 1 to 1 of 1  
Command ==> Scroll ==> PAGE
```

Version scope objects for scope "ADMR1"."SSEMMDBB INDEX"

Commands: CANCEL

Line commands:

I - Insert D - Delete R - Repeat

Sel	T	Qual	Name	Oper.
*	*		*	*
--	--	----->	----->	----
* IX ADMR1		SS12*		UPDATE
***** END OF DB2 DATA *****				

Press F3 to return to the CM Version Scopes panel. The Version scope updated message appears under the Command line at the top of the panel (Figure 9-10).

```
DB2 Admin ----- CM - Version Scopes ----- Row 1 to 1 of 1
Command ==> Scroll ==> PAGE

Version scope updated

Line commands:
  VE - Versions  SO - Version scope objects  GV - Generate new base version
  INS - Insert   U - Update  DEL - Delete  I - Details on version scope

Sel          ID Owner      Name                               Comment
      * *          *
----->
      3 ADMR1      SSEMMDBB INDEX          INDEXES ONLY FORSSEMMD
***** END OF DB2 DATA *****
```

Follow the same instructions to create a second version scope for the target indexes (Figure 9-11).

```
DB2 Admin ----- CM - Version Scopes ----- Row 1 to 2 of 2
Command ==> Scroll ==> PAGE

Line commands:
VE - Versions SO - Version scope objects GV - Generate new base version
INS - Insert U - Update DEL - Delete I - Details on version scope

Sel          ID Owner      Name                               Comment
      * *          *
-----
      3 ADMR1      SSEMMDBB INDEX                     INDEXES ONLY FORSSEMMD
      4 ADMR1      SSEMMDBC INDEX                     INDEXES ONLY FORSSEMMD
***** END OF DB2 DATA *****
```

242 Managing IBM DB2 10 for z/OS Using the IBM DB2 Administration Tool for z/OS Version 10

Now that the version scopes have been defined, we are ready to move on to Compare.

DB2 Object Comparison Tool can be accessed off of the main menu of the DB2 Administration Tool. The abbreviation can be found at the bottom of the panel and may vary depending on what was specified when the product was customized. As can be seen in Figure 9-12, the abbreviation used by our example is CP.

```
DB2 Admin ----- DB2 Administration Menu 10.1.0 ----- 20:08
Option ==> cp

    1 - DB2 system catalog                DB2 System: DBOB
    2 - Execute SQL statements            DB2 SQL ID: ADMR1
    3 - DB2 performance queries          Userid   : ADMR1
    4 - Change current SQL ID             DB2 Schema: ADMR1
    5 - Utility generation using LISTDEFS and TEMPLATES DB2 Rel   : 1015
    P - Change DB2 Admin parameters
    DD - Distributed DB2 systems
    E - Explain
    Z - DB2 system administration
    SM - Space management functions
    W - Manage work statement lists
    X - Exit DB2 Admin

    CM - Change management

                                                    More:      +

Interface to other DB2 products and offerings:

    CP - DB2 Object Comparison Tool
```

Figure 9-12 DB2 Administration Tool main menu

The steps for defining a Compare is displayed on the DB2 Object Comparison Tool Menu (Figure 9-13). Start with number 1 - Specify compare source.

```
Compare ----- DB2 Object Comparison Tool Menu ----- 20:12
Option ==> 1

    1 - Specify compare source (new)      Status:
    2 - Specify compare target (old)      Incomplete
    3 - Specify compare masks             Incomplete
    4 - Specify fields to ignore          None specified
    5 - Generate compare job              Using defaults
                                         Not generated

    W - Walk through steps 1 - 5 in sequence
    V - Generate job to extract version file from source only

    R - Reset all
    RS - Reset Source
    RT - Reset Target

    S - Save dialog
    M - Manage/Restore dialog
    MC - MultiCompare
```

Figure 9-13 DB2 Object Comparison Tool Menu

As can be seen on the Specify Compare Source panel (Figure 9-14), there is a list of four origins from which to choose. We are going to use the object definitions in the version scope, so we are going to enter VS on the Option line of the Specify Compare Source panel.

```
Compare ----- Specify Compare Source ----- 20:18
Option ==> vs

1 - Source is from a DDL file
2 - Source is from the DB2 catalog
3 - Source is from a compare version file

VS - Source is from the DB2 catalog and the objects are selected from
a version scope
```

Figure 9-14 Specify Compare Source

The Specify Source Version Scope panel prompts for the name of the version scope to be used and the data set name for the version file where the extracted definitions are to be stored for the comparison (Figure 9-15).

Note that the Owner and Name fields under the Specify version scope (Source) keyword support a lookup function as identified by the question mark.

```
Compare ----- Specify Source Version Scope -----
Command ==>

Specify version scope (Source):
  Owner . . . . . ADMR1   >      (? to look up)
  Name  . . . . . ?       >      (? to look up)

Specify compare version file output:
  Data set name . COMPARE.SOURCE.VER01

Enter a description (optional):
  Description . .
```

Figure 9-15 Specify Source Version Scope

As a result of the “lookup” function, a list of version scopes is displayed. Enter a + (plus) sign next to the name of the desired version scope (Figure 9-16).

Attention: You must press Enter for the version scope to be included in the comparison. The plus sign converts to an asterisk (*).

```
DB2 Admin ----- CM - Version Scopes ----- Row 1 to 2 of 2
Command ==>                                     Scroll ==> PAGE
Select by typing '+'
Line commands:
VE - Versions  SO - Version scope objects  GV - Generate new base version
INS - Insert   U - Update  DEL - Delete    I - Details on version scope

Sel          ID Owner      Name                               Comment
*  *          *
-----
+            3 ADMR1      SSEMMDDB INDEX                INDEXES ONLY FORSEMMD
            4 ADMR1      SSEMMDBC INDEX                INDEXES ONLY FORSEMMD
***** END OF DB2 DATA *****
```

Figure 9-16 CM Version Scope list

After the version scope has been selected, its status is noted in the DB2 Object Comparison Main Menu (Figure 9-17).

```
Compare ----- DB2 Object Comparison Tool Menu ----- 20:28
Option ==>

1 - Specify compare source (new)      Status:
2 - Specify compare target (old)      Version scope specified (DB2catalog)
3 - Specify compare masks             Incomplete
4 - Specify fields to ignore          None specified
5 - Generate compare job              Using defaults
                                      Not generated

W - Walk through steps 1 - 5 in sequence
V - Generate job to extract version file from source only

R - Reset all
RS - Reset Source
RT - Reset Target

S - Save dialog
M - Manage/Restore dialog
MC - MultiCompare
```

Figure 9-17 DB2 Object Comparison Tool Menu

The next step in defining the comparison is to specify the origin of the target definition. To accomplish this task, specify a 2 on the Option line at the top of the DB2 Object Comparison Tool Menu (Figure 9-18).

```
Compare ----- DB2 Object Comparison Tool Menu ----- 20:28
Option ==> 2

1 - Specify compare source (new)      Status:
2 - Specify compare target (old)      Version scope specified (DB2catalog)
3 - Specify compare masks             Incomplete
4 - Specify fields to ignore          None specified
5 - Generate compare job              Using defaults
                                      Not generated

W - Walk through steps 1 - 5 in sequence
V - Generate job to extract version file from source only

R - Reset all
RS - Reset Source
RT - Reset Target

S - Save dialog
M - Manage/Restore dialog
MC - MultiCompare
```

Figure 9-18 DB2 Object Comparison Tool Menu

In this example, the options for the target source will also be a version scope, so a VS is entered on the Option line of the Specify Compare Target panel (Figure 9-19).

```
Compare ----- Specify Compare Target ----- 20:32
Option ==> vs

1 - Target is from a DDL file
2 - Target is from the DB2 catalog
3 - Target is from a compare version file
4 - Target is from the DB2 catalog and the objects are automatically
   selected based on the selected source objects
VS - Target is from the DB2 catalog and the objects are selected from
a version scope
```

Figure 9-19 Specify Compare Target

The selection of the version scope works the same way as it did for the Source specification. Key in the information panel. Use the lookup function if needed (Figure 9-20).

Compare ----- Specify Target Version Scope -----
Command ==>

Specify version scope (Target):

Owner ADMR1 > (? to look up)

Name ? > (? to look up)

Specify compare version file output:

Data set name . COMPARE.TARGET.VER01

Enter a description (optional):

Description . .

Figure 9-20 Specify Target Version Scopes

Select the version scope that is to be used as the definition for the target objects using the plus (+) sign (Figure 9-21).

Note: Remember to press Enter to ensure that the version scope is included in the Compare definition.

DB2 Admin ----- CM - Version Scopes ----- Row 1 to 2 of 2
Command ==> Scroll ==> PAGE

Select by typing '+'

Line commands:

VE - Versions SO - Version scope objects GV - Generate new base version
INS - Insert U - Update DEL - Delete I - Details on version scope

Sel	ID	Owner	Name	Comment
	*	*	*	*
	3	ADMR1	SSEMMDBB INDEX	INDEXES ONLY FORSSEMMD
+	4	ADMR1	SSEMMDBC INDEX	INDEXES ONLY FORSSEMMD

***** END OF DB2 DATA *****

Figure 9-21 CM Version Scopes

Based on the naming conventions used by the objects in this comparison, a mask needs to be defined. To define the mask, enter a 3 on the Option line of the DB2 Object Comparison Tool Menu to specify compare masks (Figure 9-22).

```

Compare ----- DB2 Object Comparison Tool Menu ----- 20:39
Option ==> 3

1 - Specify compare source (new)      Status:
2 - Specify compare target (old)      Version scope specified (DB2catalog)
3 - Specify compare masks           Version scope specified (DB2catalog)
4 - Specify fields to ignore          None specified
5 - Generate compare job              Using defaults
                                      Not generated

W - Walk through steps 1 - 5 in sequence
V - Generate job to extract version file from source only

R - Reset all
RS - Reset Source
RT - Reset Target

S - Save dialog
M - Manage/Restore dialog
MC - MultiCompare

```

Figure 9-22 DB2 Object Comparison Tool Menu: Specify Compare Masks

In this example, we are going to define the mask in a table, so the Owner and Name fields need to be entered on the Specify Compare Masks panel (Figure 9-23).

```

Compare ----- Specify Compare Masks -----
Option ==>

Mask Table Entry:
  Owner . . ADMR1  >      (? to look up)
  Name  . . SS12SS13    > (? to look up)
Data Set:
  Mask DSN . .
Options:
  Edit Mask . . YES (Yes/No)

```

Figure 9-23 Specify Compare Masks

When defining a mask in a table, the user is prompted to provide the owner of the mask, the name of the mask, and a description for the mask definition. The description is optional, but it is helpful when attempting to reuse mask definitions at a future time. See Figure 9-24

```

DB2 Admin ----- CM - Insert Mask ----- 20:48
Command ==>

Owner . . . ADMR1  > (Optional, default is ADMR1, ? to lookup)
Name  . . . SS12SS13    > (Required, ? to lookup)
Comment . . MASK SS12 TO SS13                                     >

```

Figure 9-24 Insert Mask

When creating a new mask, an entry must first be inserted into the DB2 ADBCMAK repository table. When the entry is inserted, the `INSERT stmt executed` message can be seen under the Command line on the CM Insert Mask panel (Figure 9-25).

```
DB2 Admin ----- CM - Insert Mask ----- 20:52
Command ==>
INSERT stmt executed

Owner . . . ADMR1    > (Optional, default is ADMR1, ? to lookup)
Name  . . . SS12SS13      > (Required, ? to lookup)
Comment . . MASK SS12 TO SS13      >
```

Figure 9-25 Insert Mask

The next step is to define the mask options. Press F3 to display the CM Mask Lines panel (Figure 9-26).

```
DB2 Admin ----- CM - Mask Lines ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> PAGE

Mask lines for mask "ADMR1"."SS12SS13"
Commands: CANCEL
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel   Sequence Type      From      To      Oper.  T
      * *              *      *      *      *
----->----->----->----->----->----->----->
      1 ?              ?      ?
***** END OF DB2 DATA *****
```

Figure 9-26 CM Mask Lines

It is not necessary to know the mask syntax when entering masks definitions in a table. Each mask line consists of an object type, a From value, and a To value.

```

DB2 Admin ----- CM - Mask Lines ----- Row 1 from 1
Command ==> Scroll ==> PAGE

Mask lines for mask "ADMR1"."SS12SS13"
Commands: CANCEL
Line commands:
    I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel      Sequence Type      From      To      Oper.  T
          * *                *              *          *  *
-----> -----> -----

*              1 IXNAME      SS12*              SS13*              UPDATE
*****
***** END OF DB2 DATA *****

```

The Operation type on the far right hand side of the CM Mask Lines panel reflects that an UPDATE has occurred. When the mask table definition was first created, the operation was an INSERT. After values are entered, the mask is flagged as being UPDATED.

```

Compare ----- DB2 Object Comparison Tool Menu ----- 21:00
Option ==>

1 - Specify compare source (new)      Status:
2 - Specify compare target (old)      Version scope specified (DB2catalog)
3 - Specify compare masks             Version scope specified (DB2catalog)
4 - Specify fields to ignore          Mask specified
5 - Generate compare job              Using defaults
                                     Not generated

W - Walk through steps 1 - 5 in sequence
V - Generate job to extract version file from source only

R - Reset all
RS - Reset Source
RT - Reset Target

S - Save dialog
M - Manage/Restore dialog
MC - MultiCompare

```

250 Managing IBM DB2 10 for z/OS Using the IBM DB2 Administration Tool for z/OS Version 10

DB2 Object Comparison Tool supports an optional Ignore function. Ignore is not necessary in the current example, so the next step is to generate a compare job. Enter a 5 on the Option line to generate Compare jobs (Figure 9-29).

```

Compare ----- DB2 Object Comparison Tool Menu ----- 21:00
Option ==> 5

1 - Specify compare source (new)      Status:
2 - Specify compare target (old)     Version scope specified (DB2catalog)
3 - Specify compare masks            Mask specified
4 - Specify fields to ignore         Using defaults
5 - Generate compare job           Not generated

W - Walk through steps 1 - 5 in sequence
V - Generate job to extract version file from source only

R - Reset all
RS - Reset Source
RT - Reset Target

S - Save dialog
M - Manage/Restore dialog
MC - MultiCompare

```

Figure 9-29 DB2 Object Comparison Tool Menu: Generate compare job

The Generate Compare Jobs panel opens. Compare produces a set of reports indicating what the differences are and what actions would be taken to make the target look like the source. Compare may optionally generate an APPLY job, which is the job that, when executed, physically changes the target and make it look like the source. The APPLY job was not generated for this scenario.

This panel contains a number of parameters that must be entered. All of the parameters from the Options heading down pertain to the APPLY job. These parameters are shown in Table 9-1.

Table 9-1 Generate Compare Jobs parameters

Description	Option	Values or defaults	Comments
Worklist information.	Worklist name.	Any eight character name.	The value entered is used as a middle qualifier in work data sets and is also used as the default work statement list member name.

Description	Option	Values or defaults	Comments
Compare Options.	Suppress Drop of Objects.	Yes / No.	If set to Yes, any objects on the target that are not on the source are retained when recreating the target to look like the source.
	Suppress Drop of Columns.	Yes / No.	If set to Yes, any additional columns that exist on the target but not on the source are retained when recreating the target to make it look like the source.
	Suppress Adding Columns.	Yes / No.	Determines if Compare should suppress adding columns to the target.
	Run SQLID.	blank / SQLID / None.	This is the SQLID that is generated in the SET CURRENT SQLID SQL statement.
	Run Validate.	Validate / None.	Performs a consistency check. Can only be performed for the source DDL and the DB2 target catalog
Change Reporting Options.		Yes / No.	Select which reports to generate as a result of the Compare.
Data Set Information.	PDS for jobs.	48 characters.	Suffix for the PDS JCL library where the COMPARE job is generated.
	Prefix for data sets.	17 characters.	High level qualifier for data set names that are generated during the Compare process.
	Changes file data set name.	48 characters.	Data set name where the changes identified by the compare can be stored. If using Enhanced Change Management, this data set can be imported as an analyzed change ready for execution on the target subsystem.
		Member name for PDS.	
Options.	Generate Online.	Yes / No.	Should the APPLY process (the job to make the target look like the source) run online or in batch?
	Single Compare Job.	Yes / No.	
	Member name.	COMPARE.	This is the name of the job stream generated in the PDS listed in the Data Set Information of this panel. Can be any eight character name.
	Generate apply job.	Yes / No / Change.	Change represents the Delta Change file that consists only of the differences or changes between the environments being compared.

Description	Option	Values or defaults	Comments
The remainder of the parameters are only pertinent IF Generate apply job = Yes.	Generate one job.	Yes / No / Per Process.	
	Member prefix.	APPLY.	This can be changed.
	As work statement list.	Yes or No.	Store the APPLY job as a work statement list.
	Use customized utility options.	Yes or No.	If set to Yes, use the previously selected utility control card options.
	Unload method.	Unload or Parallel unload.	
	Generate templates.	Yes or No.	Use templates for data set names.
	Stop on conversion error.	Yes or No.	Stop if there are any conversion errors.
	Use DEFER Yes.	Yes or No.	
	Allow rotate parts.	Yes or No.	
Optional jobs after Reload or Alter.	Run CHECK DATA.	Yes or No.	
	Take an Image Copy	after: Reload or Alter or Both or None.	
	Run REORG / REBUILD.	Mandatory or All Relevant or None.	
	Run RUNSTATS.	after: Reload or Alter or Both or None.	
	Run REBIND.	Yes or No.	

```
File Edit_Settings Menu Utilities Compilers Test Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDIT      ADMR1.JCL.CNTL(COMPARE) - 01.00                      Columns 00001 00072
Command ==>                                                Scroll ==> CSR
***** ***** Top of Data *****
000001 //ADMR1D JOB (ADMR1D),'DB2 UTILITY',
000002 /**          RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEPNAME
000003 //              REGION=OM,NOTIFY=ADMR1,
000004 //              MSGCLASS=H,CLASS=A
000005 /**
000006 /*JOBPARM SYSAFF=*,L=999
000007 /**
000008 //              IF (RC>7) THEN
000009 //              ELSE
000010 //*****
000011 /**
000012 /** STEP EXSCPO: EXTRACT VERSION SCOPE FROM DB
000013 /**
000014 //*****ADBCVSX*****
000015 //EXSCPO    EXEC PGM=IKJEFT01,DYNAMNBR=100
000016 //STEPLIB DD DISP=SHR,DSN=GOC.V1OROMO.SGOCCLIB
000017 //          DD DISP=SHR,DSN=ADB.V1OROMO.SADBLLIB
000018 //          DD DISP=SHR,DSN=DBOBT.SDSNLOAD
000019 //          DD DISP=SHR,DSN=DBOBM.RUNLIB.LOAD
```

Submit the Compare job. Examples of the output reports are shown in Figure 9-31.

```
SDSF OUTPUT DISPLAY ADMR1D   JOB27969  DSID   123 LINE  SCREEN IMAGE PRINTED
COMMAND INPUT ===>                                     SCROLL ===> PAGE
Compare index source(ADMR1.SS12ACT1) and target(ADMR1.SS13ACT1)
  No changes to Index

Compare index source(ADMR1.SS12DNG1) and target(ADMR1.SS13DNG1)
  No changes to Index

Compare index source(ADMR1.SS12DPT1) and target(ADMR1.SS13DPT1)
  No changes to Index

Compare index source(ADMR1.SS12DPT3) and target(ADMR1.SS13DPT3)
  (D)Column(s) deleted from index
  Remaining columns will have their attributes compared
  (D)Column MGR_NO dropped
  Index ADMR1.SS13DPT3 will be dropped
  Index will be recreated

Compare index source(ADMR1.SS12EMP2) and target(ADMR1.SS13EMP2)
  (D)Column(s) deleted from index
  Remaining columns will have their attributes compared
  (D)Column MANAGER dropped
  Index ADMR1.SS13EMP2 will be dropped
  Index will be recreated
```

254 Managing IBM DB2 10 for z/OS Using the IBM DB2 Administration Tool for z/OS Version 10

The only objects in this report are indexes. Thus, using a version scope can help narrow down the amount of information displayed.

9.3 Compare: Delta change file

As stated previously, the purpose of the DB2 Object Comparison Tool is to compare two environments and optionally generate an APPLY job that, when submitted, makes the target look like the source. There are times when there are multiple targets that need to be updated. The only way to accomplish this task was to run the Compare repetitively against each of the different target environments.

A new option, “Change file data set name”, has been added to the Generate Compare Jobs parameter panel. This panel is displayed when you use the 5 - Generate compare job option on the main DB2 Object Comparison Tool’s main menu (Figure 9-32).

```

Compare ----- Generate Compare Jobs -----
Option ==>

Specify the following for DB2 Object Comparison Tool:

Prefix for data sets . . . ADMR1
Changes file data set name. DELTA.CHANGE.FILE
Member name . . . . . (if Changes file is an existing PDS)

Options:
Generate online . . . . . NO (Yes/No)
Single compare job . . . . YES (Yes/No)
Member name . . . . . COMPARE (default COMPARE )
Generate apply jobs . . . . NO (Yes, No, or (Delta) Change)
Generate one job . . . . . (Yes, No, or (Per) Process)
Member prefix . . . . . (default APPLY )
As work statement list . . (Yes/No to append to work stmt list)
Use customized util opts . (Yes/No)
Unload Method . . . . . (Unload, Parallel unload)
Generate templates. . . . NO (Yes/No)
Stop on conversion error. . (Yes/No)
Use DEFER YES . . . . . NO (Yes/No)
Allow rotate parts . . . . NO (Yes/No)

```

Figure 9-32 Generate Compare Jobs parameter panel

The “Member name” field can only be used if the change file data set has been pre-allocated and is a PDS. If the file does not exist, it is automatically allocated as sequential file. In this case, the “Member name” field is not applicable.

[illegible]

Enhanced Change Management consists of three basic processes: REGISTER, ANALYZE, and RUN. There is an optional process, IMPORT, which can be used to migrate a change to the target and store it in the change management repository as a change having a status of ANALYZE. In other words, it is ready to run on the target. This setup eliminates the need for repeating the Compare process multiple times.

g IBM DB2 10 for z/OS Using the IBM DB2 Administration Tool for z/OS Version 10

General DB2 Administration Tool functions

In this part, we provide details about using the working statement list and information about other uncommon features for data administration.

This part contains the following chapters:

- ▶ Chapter 10, “Work statement lists” on page 259
- ▶ Chapter 11, “Optional features” on page 281



Work statement lists

In this chapter, we discuss work statement lists (WSLs). A work statement list is a set of restartable instructions performing a task or tasks as defined by the DB2 Administration Tool. We examine how a WSL is defined and used for the task on hand.

This chapter contains the following topics:

- ▶ Introduction to work statement lists
- ▶ Creating work statement lists
- ▶ Accessing work statement lists
- ▶ Viewing and altering work statement lists
- ▶ Cloning work statement lists
- ▶ Interpreting work statement lists
- ▶ Validating work statement lists
- ▶ Running work statement lists
- ▶ Restarting work statement lists
- ▶ Environment variables

10.1 Introduction to work statement lists

A work statement list is a set of instructions performing a task or tasks. These tasks are varied, such as running programs, DB2 utilities, executing SQL statements, and issuing DB2 commands. The benefit of using a work statement list is that it is restartable if you have configured to use a checkpoint table.

Work statement lists are created at numerous places in DB2 Administration Tool and the user can also create one from scratch or alter one. Just about every process in DB2 Administration Tool and DB2 Object Compare allows the user to create a work statement list.

10.2 Creating work statement lists

As stated above, work statement lists can be created by just about every process in DB2 Administration Tool and DB2 Object Compare (for example, ALT, MIG, and Compare) and in fact some processes *must* use work statement lists (for example, change management). Every facility provides a YES or NO option to create work statement lists (Figure 10-1).

```
----- ALTER - Build Analyze and Apply Job -----
Option ==>

Specify the following:

Worklist information:
  Worklist name . . . . . D17676  (also used as middle qualifier in DSNs)
  Prefix for data sets . . . DBAUSER

Data set information:
  PDS final qualifiers . . . JCL.PDS
  Member name . . . . . ADBALTER
  Delete member name . . . ADBDELET (Optional job to delete work data sets)

Options:
  Generate online . . . . . NO      (Yes/No)
  Generate one job . . . . . YES     (Yes/No)
  Member name or prefix . . APPLY
  As work statement list . . YES    (Yes/No)
  Unload method . . . . . U         (Unload, Parallel unload, HPU)
  Authorization Switch ID . .      (SQLID to connect as, blank or <NONE>)

Optional processes:
  Run CHECK DATA . . . . . YES     (Yes/No)
```

Figure 10-1 Example of creating an WSL using the ALTER function

Selecting YES creates a work statement list to perform the changes contained in the result of the ALTER. The alternative is to use JCL to perform the work to be done and restartability would have to managed manually.

There are three other ways to create work statement lists. One is free form, which is described in 10.3, “Accessing work statement lists” on page 262, another is cloning an existing work statement list to create another work statement list, which is also described in 10.3, “Accessing work statement lists” on page 262, and the last is appending statements to a work statement list via the prompting facility.

When prompting is activated, one of the options is to add the statement to a work statement list (Figure 10-2).

```
DB2 Admin ----- DSNA Statement Execution Prompt ----- 09:12
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'DBAUSER.WSL.PDS'
Work statement list name ==> D17676      Action ==> A (Append or Replace)
                                           More:      +

Statement that is about to be executed (first 28 lines):
SET CURRENT SQLID = 'ADMFO01'
```

Figure 10-2 Statement execution prompt panel

You can either replace the designated work statement list with the prompted statement or append the prompted statement to the designated work statement list. If the work statement list DSN or name does not exist, it is created and the statement accordingly added.

When a work statement list DSN is created, it is created as a partitioned data set with the work statement list name as the member, which allows users to create different work statement list data sets for whatever reason they want.

Caution: It is not recommended to alter the work statement lists created by change management. To do so could cause an unexpected result.

10.3 Accessing work statement lists

If you have a work statement list data set, the next question is “How do I see it?”. To access the work statement lists, you can either select option W on the DB2 Administration Tool main menu or you can use the WSL primary command anywhere in either DB2 Administration Tool or DB2 Object Compare. After you select option W or you use the WSL primary command, you will see the primary work statement list management panel (Figure 10-3).

DB2 Admin ----- DSNA Manage Work Statement Lists ----- 17:07	
Option ==>	
1 - Show work statement list library	DB2 System: DSNA
2 - Show work statement list	DB2 SQL ID: DBAUSER
Work stmt list dsn . . . 'DBAUSER.WSL.PDS'	
Work stmt list name . . DBOC5001	
Restart User ID	

Figure 10-3 Manage Work Statement Lists panel

Once on this panel, you need to enter the work statement list DSN name that you want to manage (that is, browse, manipulate, and so on). For option 1, the only necessary item of information is the work statement list DSN. If you would like to use option 2, you need to provide the work statement list name as well as the work statement list DSN. The restart ID field indicates which user ID should be used to detect the presence of a checkpoint record. The restart user ID field is discussed in 10.9, “Restarting work statement lists” on page 272, as it is really only pertinent to restarting a work statement list.

After you provide the needed information and choose option 1, you see the Work Statement List Library contents panel (Figure 10-4).

```
DB2 Admin ----- Work Statement List Library: 'DBAUSER. Row 1 to 14 of 66
Command ==> Scroll ==> PAGE

Line commands:
S - Show R - Run in batch D - Delete C - Copy A - Append Q - Clone
I - Interpret V - Validate E - Edit O - Run online B - Checkpoint

Sel Name      Created      Changed      ID      Restart
*          *          *          *          *
-----
ALC          2010/09/03 2010/09/03 11:38 DBAUSER
ALC1         2010/10/04 2010/10/04 17:32 DBAUSER
ALC2         2010/09/03 2010/09/03 11:40 DBAUSER Y
ALTERIL      2010/10/25 2010/10/25 13:48 DBAUSER
ALTTS1       2010/03/17 2010/03/17 08:02 DBAUSER Y
AUTHSW1      2010/10/20 2010/10/20 10:35 DBAUSER
AUTHSW2      2010/10/20 2010/10/20 10:33 DBAUSER
BASEPRCA     2010/10/20 2010/10/20 07:43 DBAUSER
BASEPRCB     2010/10/27 2010/10/27 15:01 DBAUSER
BASEPRCC     2010/08/26 2010/08/26 11:25 DBAUSER
BASEPRCD     2010/03/12 2010/03/12 10:45 DBAUSER
BASEPRCE     2010/03/15 2010/03/15 08:42 DBAUSER Y
BASEPRCF     2010/10/20 2010/10/20 08:20 DBAUSER
BLOBTAB      2010/02/16 2010/02/16 03:39 DBAUSER
```

Figure 10-4 Work Statement List Library contents panel

This panel shows you what work statement lists are present in the library you provided in the prior panel along with their created and changed dates, the user ID that created the work statement list, and an indicator as to whether any restart/checkpoint records exist for that work statement list. From this panel, you can perform many operations against a work statement list, as well as view the checkpoint records associated with that work statement list (if any). You can delete and copy one work statement list and append it to another.

10.4 Viewing and altering work statement lists

There are two ways to view a work statement list:

- The show line command
- The edit line command

Use the S line command to show a work statement list. The Show Work Statement List panel opens (Figure 10-5).

```
DB2 Admin ----- Show Work Statement List: ALC1 ----- Row 1 to 14 of 843
Command ==>                                           Scroll ==> PAGE

Line commands:
D - Delete I - Insert E - Edit C - Copy M - Move A - After B - Before
R - Repeat

Select Type Statement
*      *
----->
COM -- EDITED BY DBAUSER ON 2010/06/25 AT 08:40
COM -- EDITED BY DBAUSER ON 2010/06/25 AT 08:39
COM -- START OF ADD STATEMENTS BY ADB2A2WL EXEC
COM -- BEGIN OF ADB27TUL
COM -- TABLE SPACE PARTITIONS = 0
COM -- UNLOAD H =
ADM PARALLEL UNLOAD
COM -- LOB COUNT = 0
COM -- XML COUNT = 0
ADM TSODELETE 'DBAUSER.ALC1.CNT.T001'
COM --#RESTART BOB
UTL TEMPLATE UTLPUNCH DSN DBAUSER.ALC1.CNT.T001..          DISP
UTL  TEMPLATE UTLREC DSN DBAUSER.ALC1.ULD.T001..          UNIT
COM --#RESTART 1
```

Figure 10-5 Show Work Statement List panel

This panel can give you a basic view of a work statement list but it shows a relatively unedited view of the statements and all on one line requiring the scrolling of long lines to see them all. To edit from here you can use indicated line commands to delete, insert, edit, copy, and so on. If you use the insert line command here you will see the insert work statement panel (see Figure 10-6 on page 265).

```
DB2 Admin ----- DSN Insert Work Statement ----- 17:15
Option ==>
```

Enter the type of statement you want to insert:

- 1 - A comment statement
- 2 - A definition SQL statement
- 3 - A authorization SQL statement
- 4 - A update SQL statement
- 5 - A DB2 command
- 6 - A DSN command
- 7 - A DB2 utility statement
- 8 - A DB2 Admin statement
- 9 - A restart statement

Figure 10-6 Insert work statement panel

Use the E line command to edit a work statement list. The Edit Work Statement List panel opens (Figure 10-7).

```
DB2 Admin ----- Edit Statement ----- Columns 00001 00072
Command ==> Scroll ==> PAGE
```

```
***** ***** Top of Data *****
==MSG> If any changes are made, all statements will be saved to the work
==MSG> statement list. Enter the CANCEL primary command to cancel edit without
==MSG> saving the data. Enter the SAVE primary command to save the data
==MSG> without ending the edit session.
==MSG>
000001 -- EDITED BY DBAUSER ON 2010/06/25 AT 08:40
000002 -- EDITED BY DBAUSER ON 2010/06/25 AT 08:39
000003 -- START OF ADD STATEMENTS BY ADB2A2WL EXEC
000004 -- BEGIN OF ADB27TUL
000005 -- TABLE SPACE PARTITIONS = 0
000006 -- UNLOAD H =
000007 PARALLEL UNLOAD;
000008 -- LOB COUNT = 0
000009 --#RESTART RSTR3
000010 -- XML COUNT = 0
000011 TSODELETE 'DBAUSER.ALC1.CNT.T001';
000012 --#RESTART BOB
000013 TEMPLATE UTLPUNCH DSN DBAUSER.ALC1.CNT.T001
000014 DISP (NEW,CATLG,DELETE)
000015 SPACE (30,30) TRK
```

Figure 10-7 Edit Work Statement List panel

This panel, as you can see, allows for a more formatted view of the work statement list, but it also allows you to free form edit (that is, overwrite characters, insert anything at all, and so on) and, as a result, can result in an invalid work statement list.

10.5 Cloning work statement lists

You can create a clone of a work statement list by using the Q line command on the Work Statement List Library contents panel (Figure 10-4 on page 263). The difference between a clone and a copy is that when you clone a work statement list, you can mask names of objects. For example, if you have a work statement list built to perform the changes in one subsystem, you can clone it to create the work statement list to perform the changes in another subsystem with the same name patterns but not the same names (instead of altering DBTEST1, for example, you would clone the copy to alter DBPROD1).

After you enter the Q line command to produce a clone, the Clone Work Statement List panel opens (Figure 10-8).

```
DB2 Admin ----- Clone Work Statement List -----
Command ==>                                         Scroll ==> PAGE

Input work stmt list information:                   DB2 System: DSNA
Work stmt list . . . : ALC1                        DB2 SQL ID: DBAUSER
from library . . . : 'DBAUSER.WSL.PDS'

Output work stmt list information:
Library (PDS name) . : WSL.PDS
Work stmt list . . . : AUTHSW2 (will be new PDS member)

Execution mode . . . . : TSO      (BATCH or TSO)
PDS for jobs . . . . :
PDS member . . . . :
Unit type . . . . : SYSDA

Use Masking. . . . . : NO      (Yes/No)
Apply masking to data set names. . : NO      (Yes/No=default)

Use local DB2 catalog information to replace: (Yes/No)
Authorizations . . . . . : NO
Partitioning attributes . . . . . : NO
Table space and index attributes . : NO

Additional parameters:
Message output file : ADB.DEBUG.LIST

M - Browse message output file
```

Figure 10-8 Clone Work Statement List panel

This panel allows you to specify the original (input) work statement list and choose a location for the output. You can generate the clone of a work statement list either in the foreground using the TSO execution mode or in batch. Additionally, here is where you specify the usage of masking. You can choose to use masking against the object names or the data set names. In many cases, you might want to use local DB2 catalog values in lieu of the contents of the work statement lists for items, such as authorizations, partitioning attributes, and table space and index attributes. Each of these items can be chosen here.

10.6 Interpreting work statement lists

If you want to see a view of a work statement list in the context of the objects that will be accessed, you can use the I line command at the Work Statement List Library contents panel (Figure 10-4 on page 263) to interpret a work statement list. If the interpret line command is used, the Interpret Work Statement List Options panel opens (Figure 10-9).

```
DB2 Admin ----- Interpret Work Statement List Options -----

Specify S to select Work Statement List statement types:

SQL:                                DB2 Utilities:                                DB2 Commands:                                More:      +
  S DDL                                S Load/Unload                                S Plan/packages
    ALTER                                LOAD                                BIND
    CREATE                                UNLOAD/REORG UNLOAD                                REBIND
    DROP                                S Backup/Recovery                                FREE
    COMMENT ON                                COPY                                S Other
    LABEL ON                                COPYTOCOPY                                RUN
    SET                                MERGECOPY                                START/STOP
  S DCL                                MODIFY                                Other
    GRANT                                QUIESCE                                Admin:
    REVOKE                                REBUILD                                S Data set
  S DML                                RECOVER                                ALLOC
    DELETE                                REPORT                                TSODELETE
    INSERT                                S Other                                LISTDEF
    UPDATE                                CHECK                                TEMPLATE
    Other                                DIAGNOSE                                ADBSYSIN
    COMMIT                                REORG                                S Other
    Comments                                REPAIR                                ADBPAUSE
  S Other                                RUNSTATS                                UTILFROM
    STOSPACE                                REXX Execs
    Other
```

Figure 10-9 Interpret Work Statement List Options panel

```

DB2 Admin ----- Interpret Work Statement List: ALC1 --- Row 1 to 16 of 296
Command ==> Scroll ==> PAGE

Line commands: S - Show object V - View statement

Sel   Seq Action      Object Type  Qual      Name      Note
  *     *
----->----->-----
      7 PARALLEL
     11 TSODELETE DATASET      DBAUSER.ALC1.CNT.T
     13 TEMPLATE DATASET    UTLPUNCH DBAUSER.ALC1.CNT.T
     14 TEMPLATE DATASET    UTLREC   DBAUSER.ALC1.ULD.T
     15 UNLOAD      TABLESPA
     17 ENDPARALLE
     19 ALLOC      DATASET    DD
     20 ALLOC      DATASET    DD
     22 ALLOC      DATASET    DD      DBAUSER.ALC1.IFF(T
     23 TSODELETE DATASET      DBAUSER.ALC1.CNC.T
     24 ALLOC      DATASET    DD      DBAUSER.ALC1.CNT.T
     25 ALLOC      DATASET    DD      DBAUSER.ALC1.CNC.T
     26 ADMINALTER TABLE      "DBAUSER "SALES"
     29 DROP      TABLE      DBAUSER SALES
     59 SET      SQLID      DBAUSER
     61 CREATE      TABLE      DBAUSER SALES

```

The information shown here is a form of shorthand or pseudocode version of the activities present in the work statement list. From here you can navigate to information about the object in the DB2 catalog (using the S line command) or you can see the raw work statement list statement (using the V line command).

If you have made changes to a work statement list or you just want to verify its contents, you can validate it. Use the V line command on the Work Statement List Library contents panel (Figure 10-4 on page 263). This command produces a batch job that can be submitted and the work statement list is validated for consistency. Currently, validate is a batch-only process.

When you are ready to execute the contents of a work statement list, you have two primary options to run it. You can use the R or RO line commands to execute the work statement list in batch or you can use the O or OO line commands to execute the work statement lists in the foreground. The difference between the R and RO line commands and the difference between the O and OO line commands are explained in 10.9.4, “Restart report only” on page 275.

If there are any restart records present in the checkpoint table, you are prompted to proceed by either keeping the existing checkpoint record, which means you will restart from the last point of failure, or deleting the existing checkpoint record, which means that the work statement list starts from the beginning. This panel only appears if checkpoint records are present for the work statement list (Figure 10-11).

```

DB2 Admin ----- Confirm Checkpoint Record Removal ----- 13:13

Table Owner . . : ADB10PAR

Userid . . . . : DBAUSER      Worklist . . : ALC1

A restart checkpoint record exists for this WSL.
Would you like to remove it:      (Yes/No)

```

Figure 10-11 Confirm Checkpoint Record Removal panel

After deciding whether to restart or not, press Enter and proceed.

If you have DB2 High Performance Unload installed and enabled and you have any unloads present when the work statement list runs, a panel opens and asks which type of unload you want to use for the execution (Figure 10-12).

```

DB2 Admin ----- HPU Unload Prompt ----- 12:48

Customization indicates that HP Unload is installed:
Use HPU for UNLOAD . . .      (Yes/No)

```

Figure 10-12 HPU Unload Prompt panel

This panel allows you to choose what type of unload to use for all unloads being performed in that work statement list. After the choice is made, press Enter. At this point, several panels can appear depending on the content of that work statement list and the line command used. Each of these panels are demonstrated in turn along with the conditions for each.

If you are running a work statement list in batch using either the 'R' or 'RO' line commands and the work statement list contains processes that can be run in parallel (that is, multiple unloads), you are asked whether to generate one single member or process or multiple members and processes that could be run in parallel. A panel opens and asks whether to generate one or multiple members and, if multiple members are generated, the name of the partitioned data set in which to put them (Figure 10-13).

```

DB2 Admin ----- Specify Job Parameters ----- 12:51

Enter/verify the following:
Generate one job. . . . NO      (Yes, No, or (Per) Process)
Job library PDS . . . . test.pds
Member prefix . . . . aa      (Prefix, max 2 chars)
Jobname = member? . .      (Yes/No)

```

Figure 10-13 Specify Job Parameters panel

Additionally, if multiple members are requested, the user needs to enter a member prefix for the generated members. The prefix length can vary depending on how many members are expected to be created. This information entered is used to create the members to run the work statement list in the partitioned data set specified. If only one job is generated, a temporary member is created that can be subsequently submitted. This panel does not appear if you are running the work statement list online (that is, using the O or OO line commands).

After this information is entered or, if the panel did not appear, the tool determines whether to show the Specify Restart Information panel (Figure 10-14). The criteria that determines whether this panel opens is as follows:

- The RO or OO line command was used to run the work statement list. These line commands are a variation on the original respective run line commands and force this panel to appear.
- The R or O line command was used to run the work statement list and there is a checkpoint record still present for the work statement list (that is, it was not deleted on the Confirm Checkpoint Removal panel in Figure 10-11 on page 269).

```
DB2 Admin ----- Specify Restart Information: BASEPRCB -- Row 1 to 2 of 2
Command ==>                                           Scroll ==> PAGE

Commands: CONTINUE
Line commands:
  B - Checkpoint  V - Edit Restart Info  R - Toggle Restart Report Only
  C - Toggle Ckpt Env  I - Toggle Input Env

Sel      Suffix  Restart  Ckpt Fnd  ENV Ckpt  Input  Report Only  User Restart
*        *      *      *      *      *      *      *
-----
      AA10CREA  Y      N      Y      N      N      BOB
      AA20R001  Y      N      Y      N      N
***** END OF DB2 DATA *****
```

Figure 10-14 Specify Restart Information panel

A blank suffix indicates that only a single job was generated, whereas a suffix with a value is one of many generated jobs. The Restart column indicates the method of restart to be used. Y indicates a system-generated restart using the checkpoint record, N indicates to start the job from the beginning, and U indicates a user-specified restart to the point indicated in the user restart column on this same panel. The Ckpt Fnd column indicates whether a checkpoint record exists for this particular job. If you are expecting to restart a failed job and this field is N, then there is no checkpoint record found and the job will start from the beginning.

The ENV Ckpt column and ENV Input columns are pertinent to environment variables and are discussed in detail in 10.10, “Environment variables” on page 279. The User Restart column shows a non-blank value if a user-defined restart point exists in that generated job.

This panel is primarily used only if you are restarting a work statement list and can be used to change how the work statement list members run. You can toggle where the environment variable come from in each generated job (see 10.10, “Environment variables” on page 279 for more details). You can select Toggle Restart Report Only, which generates the job and simulates a restart without actually making any changes if you are not sure or would like to confirm how the restart will be accomplished. This panel is discussed further in 10.9, “Restarting work statement lists” on page 272.

Caution: Use care when making alterations to how these work statement lists will be run, as injudicious use of this panel may cause incorrect changes when running the work statement list.

Each member to be generated is shown as a separate row on this panel. The suffix shows the member name and, if the suffix is blank, it means this is a single job execution. Each row also lists the name of the first user restart point, if any (see 10.9, “Restarting work statement lists” on page 272 for more details). If you would like to change any environment variables, you can either navigate to the checkpoint row by using the B line command and update them on the actual checkpoint record or you can use the V line command and designate an override value for certain environment variables (Figure 10-15).

When you are done on this panel, enter the CONTINUE command or move the cursor to CONTINUE and press Enter.

```

DB2 Admin ----- Edit WSL Restart Variable Overrides ----- 13:22
Command ==>

DB2 System: DSNA
DB2 SQL ID: DBAUSER

Enter/Verify:

Userid . . . . . DBAUSER
Worklist . . . . . BASEPRCB
Suffix . . . . . AA10CREA
Restart . . . . . YES
Envckpt . . . . . YES
Envinput . . . . . NO
Restart Report Only . . NO
User Restart Name . . . BOB
SQLID . . . . .
SCHEMA . . . . . >
Server . . . . .
Path . . . . . >
Precision . . . . .
Rules . . . . .
Decfloat Rounding Mode . >
Routine Version . . . . >

More: +

Press ENTER to an entry, or press END to cancel .

```

Figure 10-15 Edit WSL Restart Variable Overrides panel

On the panel, you can specify the restart parameter that controls if and how the work statement list is restarted, choose whether to use environment variables from the checkpoint record, choose whether to use environment variable from the work statement list input prior to the restart point, whether to run the restart as a report only, and choose a user restart point to start at (or not). After you are satisfied with the information on this panel, press Enter and you will be returned to the Specify Restart Information panel (Figure 10-14 on page 270).

If you chose to run the work statement list in batch (using the R or RO line commands), you now see the single JCL member with the job to execute the work statement list (if you selected generate one job) or you see the member list for the partitioned data set in which you specified to put the generated modules (if you specified generate multiple jobs).

If you chose to execute the work statement list online (using the O or OO line commands), you see the status panel showing the progress of the work statement list execution.

10.9 Restarting work statement lists

Sometimes work statement lists might not finish for whatever reason. If this event occurs, it is easy to restart the work statement list after correcting the problem. With work statement lists, there are two types of restart:

- ▶ System-controlled
- ▶ User-controlled

System-controlled restart is managed by the DB2 Administration Tool and requires no real intervention to restart the work statement list. An attempt will be made to restart the failing work statement list at the prior point of failure.

User-controlled restart allows a user to restart a work statement list in a different place than the former failing command.

In any case, restarting a work statement list is done the same way as running it in the first place (that is, there is no line command explicitly used to restart a work statement list) and you simply run the work statement list as either batch using the R or RO line commands or online using the O or OO line commands.

10.9.1 Running a work statement list and implicit commits

Restart information is managed in a set of checkpoint tables by user, work statement list name, and member/suffix (which will be blank for a single job and have a specific value other than blank for multiple jobs) and this information is updated as a work statement list is executed. Along with a count of the checkpoint (or “commits”), the value of several environment variables at each point of execution are stored (for more information about environment variables as they pertain to work statement lists, refer to 10.10, “Environment variables” on page 279).

For each work statement list implicit commit, a counter is incremented and the values of several environment variables are stored. If a failure occurs, the stored commit count and environment variables are used to restart the work statement list as the point of failure. Additionally, any DB2 commit statements result in an implicit work statement list commit as well. The execution output for a work statement list shows what the commit count is at any given point of the execution (Figure 10-16 on page 273).

```

SDSF OUTPUT DISPLAY DBAUSERD J0089265  DSID   112 LINE 27      COLUMNS 02- 81
COMMAND INPUT ==>                                SCROLL ==> CSR

ADB5044I *** INPUT STATEMENT:
-STA DB(RLSDB9) SPACE(RLSTS1) ACCESS(R0);
ADB5060I *** Implicit commit performed, number:                1

-STA DB(RLSDB9) SPACE(RLSTS1) ACCESS(R0)

DSN9022I #DSNA- DSNTDDIS 'START DATABASE' NORMAL COMPLETION

ADB5060I *** Implicit commit performed, number:                2
ADB5044I *** INPUT STATEMENT:
  TSODELETE 'DBAUSER.DSNA.PM22596.CNT.T0001';
*** TSODELETE Successful
ADB5044I *** INPUT STATEMENT:
  TSODELETE 'DBAUSER.DSNA.PM22596.ULD.T0001';
*** TSODELETE Successful
ADB5044I *** INPUT STATEMENT:
  TEMPLATE UTLPUNCH DSN 'DBAUSER.DSNA.PM22596.CNT.T0001'
    UNIT SYSDA;

ADB5044I *** INPUT STATEMENT:
  TEMPLATE UTLREC DSN 'DBAUSER.DSNA.PM22596.ULD.T0001'
    UNIT SYSDA;

ADB5044I *** INPUT STATEMENT:
  UNLOAD TABLESPACE RLSDB9.RLSTS1
  FROM TABLE
  "DBAUSER"."DEFINETB1"
  UNLDDN UTLREC
  PUNCHDDN UTLPUNCH;
ADB5060I *** Implicit commit performed, number:                3
DSNU000I   357 08:06:34.79 DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = PM2259
DSNU1044I   357 08:06:34.84 DSNUGTIS - PROCESSING SYSIN AS EBCDIC
DSNU050I   357 08:06:34.85 DSNUGUTC - TEMPLATE UTLPUNCH DSN 'DBAUSER.DSNA.PM22
DSNU1035I   357 08:06:34.85 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I   357 08:06:34.85 DSNUGUTC - TEMPLATE UTLREC DSN 'DBAUSER.DSNA.PM2259
DSNU1035I   357 08:06:34.85 DSNUJTDR - TEMPLATE STATEMENT PROCESSED SUCCESSFULLY
DSNU050I   357 08:06:34.85 DSNUGUTC - UNLOAD TABLESPACE RLSDB9.RLSTS1
DSNU650I #DSNA- 357 08:06:34.85 DSNUUGMS - FROM TABLE "DBAUSER"."DEFINETB1" UN
DSNU1038I   357 08:06:34.89 DSNUGDYN - DATASET ALLOCATED. TEMPLATE=UTLREC
                                DDNAME=SYS00006
                                DSN=DBAUSER.DSNA.PM22596.ULD.T0001
DSNU1038I   357 08:06:34.93 DSNUGDYN - DATASET ALLOCATED. TEMPLATE=UTLPUNCH
                                DDNAME=SYS00007
                                DSN=DBAUSER.DSNA.PM22596.CNT.T0001
DSNU253I   357 08:06:34.95 DSNUUNLD - UNLOAD PHASE STATISTICS - NUMBER OF RECOR
DBAUSER.DEFINETB1
DSNU252I   357 08:06:34.95 DSNUUNLD - UNLOAD PHASE STATISTICS - NUMBER OF RECOR
RLSDB9.RLSTS1
DSNU250I   357 08:06:34.95 DSNUUNLD - UNLOAD PHASE COMPLETE, ELAPSED TIME=00:00
DSNU010I   357 08:06:34.96 DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETUR
ADB5060I *** Implicit commit performed, number:                4

```

Figure 10-16 Work Statement List Execution Output

You can see in this report that lines beginning with a message code of ADB5060I indicate the commit number.

10.9.2 System-controlled restart

As stated before, system-controlled restart simply means we want to attempt to restart the work statement list at the failing instruction. The checkpoint record for the work statement list is read, the environment variables are set to their values at the point of failure, and the work statement list execution resumes at the prior point of failure.

To restart a work statement list, run it using your desired method (batch or online) and take care not to delete any existing checkpoint records. If you delete the checkpoint record, the work statement list starts from the beginning. When you get to the Specify Restart panel (Figure 10-14 on page 270), confirm that the Restart column for the jobs to be restarted has a value of Y and that the Ckpt column also has a Y. If either of these columns is set to N, then that job starts from the beginning if it is submitted. If the panel contents are as expected, you can simply use the CONTINUE primary command or move your cursor to CONTINUE and press Enter. The jobs is generated and can be submitted.

One important item of information to note regarding restart is that checkpoint information is by user, so if you want to restart a work statement list that was submitted by another user and subsequently failed, you need to specify the restart user ID on the Manage Work Statement List panel (Figure 10-1 on page 260). If you want to restart another user's work, you simply specify that user's ID here so that the correct checkpoint record is retrieved.

10.9.3 User-controlled restart

Perhaps there is a situation where you want to restart a work statement list at a different point than where it originally failed. This task can be easily accomplished by adding user-defined restart points. A user-defined restart point is simply a marker added to a work statement list with an identifier of your own choosing. The user-defined restart points can be added either by showing the work statement list and inserting a restart record or by performing a free form edit and adding the restart point manually (see 10.4, "Viewing and altering work statement lists" on page 264). If you want to add a user-defined restart point, the identifying string on the insert work statement panel (Figure 10-6 on page 265) can be anything that is not the specific values of YES, NO, FORCE, or a pure numeric value (for example, you cannot use 100, but you can use A100) because these values are reserved. If you want to add a user-defined restart point using free form editing through the Edit Work Statement List panel (see 10.4, "Viewing and altering work statement lists" on page 264), add a line where you want to restart using the syntax `--#RESTART <string>`, where string is simply an identifier of your own choosing (again, avoid using YES, NO, FORCE, or a pure numeric value). These restart statements do nothing until you are ready to restart a work statement list. You can add as many user-defined restart points to a work statement list as you want, but only one is used for restart. The first user-defined restart record matching the specified string is used as the point of restart.

Again, to restart a work statement list, you run it using your desired method (batch or online). Because the point of restart is user-specified, the restart record is not necessary to determine the point of restart and, therefore, can be deleted if so desired. You may choose to keep the checkpoint record, as you might want to use the environment variables contained in it (see 10.10, "Environment variables" on page 279 for more information).

When you get to the Specify Restart Information panel (Figure 10-14 on page 270), you need to set the restart to U (for user) and confirm a user restart point has been found (indicated by a non-blank value in the user restart column). To accomplish this task, use the V line command on the Specify Restart Information panel to show the Edit WSL Restart Overrides panel (Figure 10-15 on page 271). Overtyping the restart value with U or type USER and confirm there is a value in the user restart field as well. After the appropriate changes are made, you can use the CONTINUE primary command or move your cursor to CONTINUE and press Enter. The jobs are generated and can be submitted.

10.9.4 Restart report only

If you would like to simulate a restart without actually performing the restart itself, you can see the results using a restart report only. To use this feature, follow the instructions for your restart of choice (see 10.9.2, “System-controlled restart” on page 274 and 10.9.3, “User-controlled restart” on page 274) to choose your restart options, but on the Specify Restart Information panel (Figure 10-14 on page 270), use the R line command to toggle the Toggle Restart Report Only option on and off. If the job is generated and submitted, it shows you where the restart will take place and the environment variables settings at the point of restart. If you use the R or O line commands to run your work statement list, it is possible that the Specify Restart Information panel might not appear. You can force the Specify Restart Information panel to appear by using the RO and OO line command for a batch or online restart, respectively.

10.9.5 An example of restarting a work statement list

The following section gives an example of running a work statement list involving a system-defined restart. The first step to running a work statement list is to navigate to the Manage Work Statement Lists panel using the WSL primary command (Figure 10-17).

```
DB2 Admin ----- DB2 Administration Menu 10.1.0 ----- 08:10
Option ==> wsl

    1 - DB2 system catalog                DB2 System: DSNA
    2 - Execute SQL statements            DB2 SQL ID: DBAUSER
    3 - DB2 performance queries          Userid   : DBAUSER
    4 - Change current SQL ID             DB2 Schema: DBAUSER
    5 - Utility generation using LISTDEFs and TEMPLATES DB2 Rel   : 1015
    P - Change DB2 Admin parameters
    DD - Distributed DB2 systems
    E - Explain
    Z - DB2 system administration
    SM - Space management functions
    W - Manage work statement lists
    X - Exit DB2 Admin
    CC - DB2 catalog copy version maintenance
    CM - Change management

More:      +

Interface to other DB2 products and offerings:
    I  DB2I
    C  DB2 Object Comparison Tool
```

Figure 10-17 Use the WSL primary command

After this primary command is entered (which can be entered anywhere in the product), the Manage Work Statement Lists panel opens. On this panel, browse to the partition data set containing your work statement lists by using option 1 (Figure 10-18). You use option 1 because you cannot run a work statement list directly from the Show Work Statement List panel, which is where option 2 would direct you.

```
DB2 Admin ----- DSNB Manage Work Statement Lists ----- 08:13
Option ==> 1

    1 - Show work statement list library          DB2 System: DSNB
    2 - Show work statement list                 DB2 SQL ID: DBAUSER

Work stmt list dsn . . . 'DBAUSER.WSL.PDS'
Work stmt list name . . D17917

Restart User ID . . . .
```

Figure 10-18 Choose option 1 to show a WSL library

After choosing the 1- Show work statement list library option and entering the partitioned data set name of the WSL library, the Work Statement List Library panel opens. On this panel, you see the work statement list you want to run. For this example, BASEPRCA is the work statement list you run. You can already see there is one or more checkpoint records present for this work statement list. Remember that a work statement list can be run as multiple jobs, each of which can create a checkpoint record. Run this work statement list in batch using the R line command (Figure 10-19).

```
DB2 Admin ----- Work Statement List Library: 'DBAUSER. Row 1 to 14 of 79
Command ==>                                     Scroll ==> PAGE

Line commands:
S - Show  R - Run in batch  D - Delete  C - Copy  A - Append  Q - Clone
I - Interpret  V - Validate  E - Edit  O - Run online  B - Checkpoint

Sel Name      Created      Changed      ID      Restart
*           *           *           *           *
-----
   ALC        2010/09/03 2010/09/03 11:38 DBAUSER
  ALCCOPY     2010/12/05 2010/12/05 07:40 DBAUSER
   ALC1       2010/11/03 2010/11/03 17:17 DBAUSER  Y
   ALC2       2010/09/03 2010/09/03 11:40 DBAUSER
  ALTERIL    2010/10/25 2010/10/25 13:48 DBAUSER
  ALTTS1     2010/11/11 2010/11/11 11:46 DBAUSER
  AUTHSW1    2010/10/20 2010/10/20 10:35 DBAUSER
  AUTHSW2    2010/10/20 2010/10/20 10:33 DBAUSER
r  BASEPRCA  2010/10/20 2010/12/30 08:19 DBAUSER  Y
  BASEPRCB   2010/10/27 2010/10/27 15:01 DBAUSER
  BASEPRCC   2010/08/26 2010/08/26 11:25 DBAUSER
  BASEPRCD   2010/03/12 2010/03/12 10:45 DBAUSER
  BASEPRCE   2010/03/15 2010/03/15 08:42 DBAUSER
  BASEPRCF   2010/10/20 2010/10/20 08:20 DBAUSER
```

Figure 10-19 Run the BASEPRCA work statement list in batch

Because there is one or more checkpoint records associated with this work statement list, you are prompted with a panel asking if you would like to remove all of the checkpoint records for this work statement list. Because you want to restart this work statement list, answer N to this prompt (Figure 10-20). Had you removed the work statement list, the work statement list jobs would all run from the beginning.

```
DB2 Admin ----- Confirm Checkpoint Record Removal ----- 08:23

Table Owner . . : ADB1OPAR

Userid . . . . : DBAUSER      Worklist . . : BASEPRCA

A restart checkpoint record exists for this WSL.
Would you like to remove it: n   (Yes/No)
```

Figure 10-20 Specify N to keep the existing checkpoint records

After you specify N to keep the existing checkpoint records, and because there are one or more unloads in the work statement list being run, the HPU Unload Prompt panel opens, which allows the user to specify whether DB2 Unload or High Performance Unload (HPU) will be used. This panel only appears if there are unloads present in the work statement list *and* High Performance Unload is installed and enabled for use in DB2 Administration Tool. Use DB2 Unload and answer N at this prompt (Figure 10-21).

```
DB2 Admin ----- HPU Unload Prompt ----- 08:25

Customization indicates that HP Unload is installed:
Use HPU for UNLOAD . . . n   (Yes/No)
```

Figure 10-21 Use DB2 Unload by answering N

After you answer the question as to what kind of unload is used, you might be presented with the Specify Job Parameters panel. This panel only appears if there are processes that can be run in parallel in the work statement list (that is, multiple unloads and so on) or processes that can be broken into multiple jobs, such as an unload, a drop, and a create. In this example, you want to generate multiple jobs, so specify NO for “Generate one job”. Because you want multiple jobs, you need to specify a job library partitioned data set in which to put the generated jobs. If the data set does not exist, it will be created. Also, because you specified multiple jobs, you must specify a member prefix that will be used. For this example, the prefix of AA is used (Figure 10-22).

```
DB2 Admin ----- Specify Job Parameters ----- 08:26

Enter/verify the following:
Generate one job. . . . NO      (Yes,No, or (Per) Process)
Job library PDS . . . . baseprca.pds
Member prefix . . . . aa      (Prefix, max 2 chars)
Jobname = member? . .      (Yes/No)
```

Figure 10-22 Specify Job Parameters example

After this information is entered and the Enter key is pressed, the Specify Restart Information panel opens. This panel only shows if there are any checkpoints present for the work statement list being run or the line commands RO or OO were used to run the work statement list. Because there are checkpoint records present, this panel opens (Figure 10-23).

```

DB2 Admin ----- Specify Restart Information: BASEPRCA -- Row 1 to 4 of 4
Command ==>                                           Scroll ==> PAGE

Commands: CONTINUE
Line commands:
  B - Checkpoint  V - Edit Restart Info  R - Toggle Restart Report Only
  C - Toggle Ckpt Env  I - Toggle Input Env

Sel  Suffix  Restart  Ckpt Fnd  ENV Ckpt  ENV Input  Report Only  User Restart
      *      *      *      *      *      *      *      *
-----
      AA10U001 Y      N      Y      N      N
      AA20DROP Y      N      Y      N      N
      AA30CREA Y      Y      Y      N      N
      AA40R001 Y      N      Y      N      N
***** END OF DB2 DATA *****

```

Figure 10-23 Example of Specify Restart Information panel

This panel shows information about each of the generated jobs/members. You can see that all of the generated jobs use RESTART(YES) as the input parameter. Keep in mind that this means that a restart occurs if there is an associated checkpoint record present. You can determine if such a record is present by reviewing the contents of the Ckpt Fnd column, which indicates whether a checkpoint record is present for the user ID, work statement list, and suffix/member. Notice that only the AA30CREA member has a checkpoint record, so this job is the only one that restarts in the middle if run. The ENV Ckpt and ENV Input values designate whether environment variables from the checkpoint record and in the work statement list prior to the restart point are used, respectively. The Report Only column is initially set to N. The User Restart column shows a value if any user-defined restart point is present in the work statement list. If there is more than one user-defined restart record, the name of the first one would be shown. Because you want to restart the job without changing any environment variables, use the CONTINUE primary command to proceed to job generation and submit the appropriate jobs.

Assume that you want to generate only a single job. Go back to the step where we specified the job parameters (Figure 10-22 on page 277) and instead of entering NO for “Generate one job”, specify YES (Figure 10-24). Because you chose YES, this is all that is required on that panel. Press Enter.

```

DB2 Admin ----- Specify Job Parameters ----- 08:36

Enter/verify the following:
Generate one job. . . . yes      (Yes,No, or (Per) Process)
Job library PDS . . . .
Member prefix . . . . BASE      (Prefix, max 2 chars)
Jobname = member? . . . .      (Yes/No)

```

Figure 10-24 Alternate Specify Job Parameters example

The generated job appears. Why did you not see the Specify Restart Information panel as you did in the original example? The reason is that in the first example, you generated multiple jobs, each with their own suffix. If you generate a single job, the suffix used is blank and there are no checkpoint records for a blank suffix. If you want to force the Specify Restart Information panel to open, run this work statement list using the RO or OO line command (Figure 10-25). Notice that the suffix is blank.

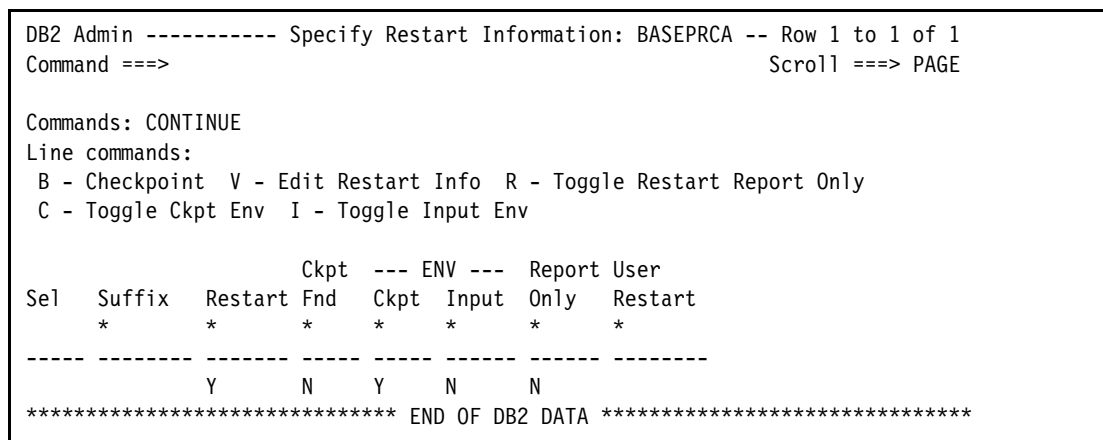


Figure 10-25 Alternate Specify Restart Information panel

10.10 Environment variables

The last element to consider for restart are the environment variables used at a particular point of restart. For a system-defined restart, you would typically not make changes to the environment variables for a restart. As a user, you can control from where the environment variables for a work statement list execution are sourced. Environment variables can come from four different sources where work statement lists are concerned:

- ▶ Initial values: These are the settings of the environment variables in DB2 upon starting a work statement list.
- ▶ Checkpoint record: Certain environment variables are stored on the checkpoint record (see 10.10.1, “Stored environment variables” on page 280).
- ▶ Work statement list input: This input can be specified to perform DB2 SET statements that precede the point of restart to reestablish values that were not stored in the checkpoint record.
- ▶ Overridden environment values: On the Edit WSL Restart Variable Override panel (Figure 10-15 on page 271), you can force environment variables to be set to a particular value at the point of restart.

When a work statement list starts, you have the initial values already in place. The work statement list is processed up to the point of restart. If you have specified that you want to use environment variables from the work statement list input by toggling Inpt Env on the Specify Restart Information panel (Figure 10-14 on page 270) or by specifying YES in the Envinput field on the Edit WSL Restart Variable Overrides panel (Figure 10-15 on page 271), this means that any SET statements prior to the point of restart are executed. This can be useful if any environment or host variables are set in the work statement list that are not stored in the checkpoint record or available to be overridden.

When the point of restart is reached, the environment variables stored in the checkpoint record are set to their values if the Env Ckpt value is set to YES (which is the default). This can be turned off, if desired, in the same fashion as the Env Inpt value. Lastly, the environment variables are set to their override values if any were specified on the Edit WSL Restart Variable Overrides panel.

10.10.1 Stored environment variables

Only certain environment variables are stored on the checkpoint records. These are also the only environment variable that are available to be overridden. They are:

- ▶ CURRENT SQLID
- ▶ CURRENT SERVER (Although technically not a DB2 environment variable, it is managed as though it were one.)
- ▶ PATH
- ▶ SCHEMA
- ▶ SESSION TIMEZONE
- ▶ CURRENT EXPLAIN MODE
- ▶ CURRENT DECFLOAT ROUNDING MODE
- ▶ CURRENT PRECISION
- ▶ CURRENT ROUTINE VERSION
- ▶ CURRENT RULES

If you include any variables using SET statements in a work statement list that you would like to have set to their values at the point of restart, you can use the Env Inpt facility to have SET statements prior to the point of restart rerun.



Optional features

In this chapter, we look at additional features and functions of DB2 Administration Tool that are optional but provide value to users. We also look at commands that can reduce the time taken to achieve goals.

This chapter contains the following topics:

- ▶ Connecting to a different DB2
- ▶ Useful DB2 Administration Tool commands
- ▶ Features you may have missed

11.1 Connecting to a different DB2

There are two methods of switching connections between DB2 subsystems when you are using the DB2 Administration Tool, depending upon whether the subsystem is remote or local. To connect to a local subsystem, use the SSID command, and for remote connections, use the CONNECT command.

11.1.1 CONNECT

This command is used to connect to subsystems that are defined in the communications database. To see which systems that you can connect to using this command, use option DD on the main menu. The Distributed DB2 Systems panel opens (Figure 11-1).

```
DB2 Admin ----- Distributed DB2 Systems ----- Row 1 to 4 of 4
Command ==> Scroll ==> CSR

Select the location you wish to use:                                DB2 System: DSNT
                                                                    DB2 SQL ID: ADMR3

Line commands:
  S - Use DDF to access remote catalog  CO - Connect to remote subsystem
  DIS - Display threads for remote system
Select Location
  *
-----
      TSTDB10
      TSTDB11
      TSTDB12
***** END OF DB2 DATA*****
```

Figure 11-1 Displaying a locations database

To connect to a remote system, for example, TSTDB11, either run CONNECT TSTDB11 or the CO line command (Figure 11-2). Both commands are shown in one panel for the sake of brevity.

```
DB2 Admin ----- Distributed DB2 Systems ----- Row 1 to 4 of 4
Command ==> connect TSTDB11 Scroll ==> CSR

Select the location you wish to use:                                DB2 System: DSNT
                                                                    DB2 SQL ID: ADMR3

Line commands:
  S - Use DDF to access remote catalog  CO - Connect to remote subsystem
  DIS - Display threads for remote system
Select Location
  *
-----
co      TSTDB10
      TSTDB11
      TSTDB12
***** END OF DB2 DATA*****
```

Figure 11-2 Connecting to a remote subsystem

After pressing Enter, you are connected to the target subsystem (Figure 11-3). We pressed PF1 to show the help message associated with the short message; this message shows the details of the subsystem to which you are connected. While connected to this subsystem, you have limited functionality within the connected subsystem as follows:

- ▶ When using the distributed DB2 systems function to access a remote DB2 system catalog, some functions in the DB2 Administration system catalog dialog are disabled. For example, you cannot issue DB2 DISPLAY or GEN commands, and unless prompting is on, you also cannot issue DB2 BIND, REBIND, or FREE commands.
- ▶ If you connect to a remote subsystem that does not have an entry in the ADB2DB2D customization table, alter, migrate, and utility jobs are not allowed, and an error message is displayed.
- ▶ To use copies of the system catalog of a remote subsystem, the local subsystem customization must specify the owner of the catalog copy version table.
- ▶ You cannot use option 1 of the Space Management function (the display page sets space by database).
- ▶ You cannot issue SM line commands on the database and table space panels.
- ▶ You cannot interface to other DB2 products from a remote subsystem

```
DB2 Admin ----- DB2 Administration Menu 7.2.0 ----- 21:04
Option ==>
Connection established
    1 - DB2 system catalog                DB2 System: DB11
    2 - Execute SQL statements            DB2 SQL ID: ADMR3
    3 - DB2 performance queries          Userid      : ADMR3
    4 - Change current SQL ID             DB2 Rel     : 815
    5 - Utility generation using LISTDEfs and TEMPLATEs
    P - Change DB2 Admin parameters
    DD - Distributed DB2 systems
    E - Explain
    Z - DB2 system administration
    SM - Space management functions
    W - Manage work statement lists
    X - Exit DB2 Admin
    CC - DB2 catalog copy version maintenance
    CM - Change management

EsaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaN
e Now connected to server TSTDB11.Product: DB2 level: 815 e
DsaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaM
```

Figure 11-3 Connected to a remote subsystem

11.1.2 SSID

SSID is a DB2 Administration Tool primary command that is used to switch between locally defined DB2 subsystems. To activate this command you need to set the :ADB2CUST variable, :SWSSID, when customizing the product. This variable can be set for each subsystem to allow greater granularity. If this variable is not set, then you will get the message *Invalid command* when you try to use the command. If the variable is set, you are switched to the main menu of the target subsystem (Figure 11-4). Note that the full menu is available.

```
DB2 Admin ----- DB2 Administration Menu 10.1.0 ----- 15:58
Option ==>

    1 - DB2 system catalog                      DB2 System: DB12
    2 - Execute SQL statements                  DB2 SQL ID: ADMR3
    3 - DB2 performance queries                Userid   : ADMR3
    4 - Change current SQL ID                  DB2 Schema: ADMR3
    5 - Utility generation using LISTDEFs and  DB2 Rel   : 915
        TEMPLATES
    P - Change DB2 Admin parameters
    DD - Distributed DB2 systems
    E - Explain
    Z - DB2 system administration
    SM - Space management functions
    W - Manage work statement lists
    X - Exit DB2 Admin
    CC - DB2 catalog copy version maintenance
    CM - Change management

                                                    More:      +

Interface to other DB2 products and offerings:
    I  DB2I
    C  DB2 Object Comparison Tool
```

Figure 11-4 Local connection menu

11.2 Useful DB2 Administration Tool commands

Here we look at some of the DB2 Administration Tool primary commands that can make you more productive and aid you in finding information.

11.2.1 ALL

The ALL command allows you to display all related objects, of a specified type, to all the objects that are currently displayed on your panel. If you have a list of all the tables that exist that were created by ADMR3 and want to get all indexes defined on those tables, then you would have to issue an X command against every table and then press Enter. The indexes related to the first table would then be displayed and, after pressing End, the next set of indexes relating to the second table on your list would be displayed. This process continues until the list of tables ends. Using the ALL X command, you can show all the indexes that are related to all the tables that are on your list in a single panel, reducing the number of keys that you must press and allowing you to act upon the whole set of indexes in one action rather than in several.

Figure 11-5 shows a list of tables with creator equal to ADMR3, and the primary command “ALL X - show me all indexes that are defined on this list of tables”.

DB2 Admin ----- DBOB Tables, Views, and Aliases ----- Row 1 from 16

Command ==> all X

Scroll ==> PAGE

Commands: GRANT MIG ALL

Line commands:

C - Columns A - Auth L - List X - Indexes S - Table space D - Database

V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping

? - Show all line commands

Se1	Name	Schema	T DB Name	TS Name	Cols	Rows	Chks	C
	*	*	* *	*	*	*	* *	*
----->-----								
	EMP	ADMR3	T ADMR3ADB	TESTING	14	-1	2	
	TD76TB01_DEPT	ADMR3	T ADMR3ADB	TD76TS01	5	-1	0	
	TD76TB02_EMP	ADMR3	T ADMR3ADB	TD76TS02	14	-1	2	
	TD76TB03_ACT	ADMR3	T ADMR3ADB	TD76TS03	3	-1	0	
	TD76TB04_PROJ	ADMR3	T ADMR3ADB	TD76TS04	8	-1	0	
	TD76TB05_PROJACT	ADMR3	T ADMR3ADB	TD76TS05	5	-1	0	
	TD76TB06_EMPPROJAC	ADMR3	T ADMR3ADB	TD76TS06	6	-1	0	
	TD76TB07_EACT	ADMR3	T ADMR3ADB	TD76TS07	5	-1	0	
	TD76TB08_EDEPT	ADMR3	T ADMR3ADB	TD76TS08	7	-1	0	
	TD76TB09_EEPA	ADMR3	T ADMR3ADB	TD76TS09	8	-1	0	
	TD76TB10_EPROJ	ADMR3	T ADMR3ADB	TD76TS10	10	-1	0	
	TD76TB11_EPROJACT	ADMR3	T ADMR3ADB	TD76TS11	7	-1	0	

Figure 11-5 ALL X command

The result of the ALL X command is shown in Figure 11-6. From this panel, you can issue one command to run against the complete list (for example, UTIL) rather than issuing the command multiple times as each index panel opens.

```
DB2 Admin ----- DB0B Indexes ----- Row 1 to 11 of 20
Command ==> Scroll ==> PAGE

Commands: DIS STA STO ALL
Line commands:
T - Tables D - Database G - Storage group P - Plans C - Columns
DIS - Display index space STA - Start index space STO - Stop index space
? - Show all line commands
```

Select	Index Name	Index Schema	Table Name	Table Schema	U	Cols	C	C	C	C
*	*	*	*	*	*	*	G	D	L	M
							*	*	*	*
----->-----										
	TD76XA01	ADMR3	TD76TB01_DEPT	ADMR3	P	1	N	N	N	N
	TD76XA02	ADMR3	TD76TB02_EMP	ADMR3	P	1	Y	Y	N	N
	TD76XA03	ADMR3	TD76TB03_ACT	ADMR3	P	1	N	N	N	N
	TD76XA04	ADMR3	TD76TB04_PROJ	ADMR3	P	1	N	N	N	N
	TD76XA05	ADMR3	TD76TB05_PROJACT	ADMR3	P	3	N	N	N	N
	TD76XA06	ADMR3	TD76TB06_EMPPROJAC	ADMR3	U	4	N	N	N	N
	TD76XA13	ADMR3	TD76TB13_PARTS	ADMR3	D	1	N	N	N	N
	TD76XA14	ADMR3	TD76TB14_DEPT	ADMR3	P	1	N	N	N	N
	TD76XB01	ADMR3	TD76TB01_DEPT	ADMR3	D	1	N	N	N	N
	TD76XB02	ADMR3	TD76TB02_EMP	ADMR3	D	1	N	N	N	N
	TD76XB03	ADMR3	TD76TB03_ACT	ADMR3	U	1	N	N	N	N

Figure 11-6 ALL X command result

The ALL command can be issued for the objects shown in Table 11-1. (However, you cannot issue ALL T from a list of tables.)

Table 11-1 Valid ALL objects

Command	Description
ALL S	All table spaces
ALL T	All tables
ALL K	All packages
ALL X	All indexes
All A	All aliases
ALL V	All first level views
All VV	All views (inc views on views)

11.2.2 Report

Report (REP) is a simple primary, or line, command that executes a batch job that produces a printable report of the selected and dependant objects in the DB2 catalog. The report has various options and only includes the options suitable for the object(s) being reported.

When you issue the REP command (Figure 11-7), the panel shown in Figure 11-8 opens. In this panel, you can select the dependant objects you want to report upon.

```
DB2 Admin ----- DB0B Databases ----- Row 1 to 1 of 1
Command ==> rep                               Scroll ==> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL
Line commands:
  T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
  DIS - Display database  STA - Start database  STO - Stop database  A - Auth
  ? - Show all line commands

Select Name      Owner      Storage      Buffer      Created      Index
      *          *          Group          Pool          DBID By      T E BPool      I
-----
      ADMR3ADB  ADMR3      DSN8G100  BP0          402 ADMR3      E BP0      N
*****
***** END OF DB2 DATA *****
```

Figure 11-7 Issuing the REP command

```
DB2 Admin ----- DB0B Generate Report from DB2 Catalog -----18:48
Option ==>

Generate batch report for all displayed databases      DB2 System: DB0B
                                                       DB2 SQL ID: ADMR3
                                                       More:      +

Object types to be included from the DB2 catalog:
Database . . . . . Y (Y,N)
Table space . . . . . Y (Y,N)
Table . . . . . Y (Y,N)
View . . . . . Y (Y,N)
Index . . . . . Y (Y,N)
Synonym . . . . . Y (Y,N)
Alias . . . . . Y (Y,N)
Trigger . . . . . Y (Y,N,D)
Storage group . . . . . N (Y,N)
Plan/package . . . . . Y (Y,N)

Include column data . . . . Y (Y,N)

Output file:
Data set name . . . . . >
Data set disposition . . OLD (OLD, SHR, or MOD)
```

Figure 11-8 Selecting the REP options

You are presented with the JCL for submission. Submit the JCL to generate a report (Example 11-1). (The report is truncated to show examples of the report for each object in the database)

Example 11-1 Report output

5697-L90 IBM DB2 Administration Tool for z/OS Database Report PAGE 2

OName	Owner	Storage Group	Buffer Pool	DBID	Created By	T E Index Buffer Pool
ADMR3ADB	ADMR3	DSN8G100	BPO	402	ADMR3	E BPO

15697-L90 IBM DB2 Administration Tool for Z/OS Table Space Report PAGE 3

OName	DB Name	Parts	Bpool	L E S I C	Tables	Act. pages	Segsz	T L
TD76TS01	ADMR3ADB	0	BPO	P N A N N	1	-1.0000000	4	Y
TD76TS02	ADMR3ADB	5	BPO	P N A N N	1	-1.0000000	0	Y
TD76TS03	ADMR3ADB	0	BPO	R N A N N	1	-1.0000000	4	Y
TD76TS04	ADMR3ADB	0	BPO	R N A N N	1	-1.0000000	4	Y
TD76TS05	ADMR3ADB	0	BPO	R N A N N	1	-1.0000000	4	Y
TD76TS06	ADMR3ADB	0	BPO	R N A N N	1	-1.0000000	4	Y
TD76TS07	ADMR3ADB	0	BPO	P N A N N	1	-1.0000000	4	Y
TD76TS08	ADMR3ADB	0	BPO	P N A N N	1	-1.0000000	4	Y
TD76TS09	ADMR3ADB	0	BPO	P N A N N	1	-1.0000000	4	Y
TD76TS10	ADMR3ADB	0	BPO	P N A N N	1	-1.0000000	4	Y
TD76TS11	ADMR3ADB	0	BPO	P N A N N	1	-1.0000000	4	Y
TD76TS12	ADMR3ADB	0	BPO	P N A N N	1	-1.0000000	4	Y
TD76TS13	ADMR3ADB	0	BPO	P N A N N	1	-1.0000000	4	Y
TD76TS14	ADMR3ADB	2	BPO	P N A N N	1	-1.0000000	4 G	Y
TESTING	ADMR3ADB	5	BPO	P N A N N	1	-1.0000000	0	Y

15697-L90 IBM DB2 Administration Tool for Z/OS Table Report PAGE 4

OName	Owner	T DB Name	TS Name	Cols	Rows Ch
OEMP	ADMR3	ADMR3ADB	TESTING	14	-1.000000000000000E+00
0					
	Column Name	Col No	Col Type	Length	Scale Null Def FP
	EMPNO	1	CHAR	6	0 N N N -1.000000000000000E+00
	FIRSTNME	2	VARCHAR	12	0 N N N -1.000000000000000E+00
	MIDINIT	3	CHAR	1	0 N N N -1.000000000000000E+00
	LASTNAME	4	VARCHAR	15	0 N N N -1.000000000000000E+00
	WORKDEPT	5	CHAR	3	0 Y Y N -1.000000000000000E+00
	PHONENO	6	CHAR	4	0 Y Y N -1.000000000000000E+00
	HIREDATE	7	DATE	4	0 Y Y N -1.000000000000000E+00
	JOB	8	CHAR	8	0 Y Y N -1.000000000000000E+00
	EDLEVEL	9	SMALLINT	2	0 Y Y N -1.000000000000000E+00
	SEX	10	CHAR	1	0 Y Y N -1.000000000000000E+00
	BIRTHDATE	11	DATE	4	0 Y Y N -1.000000000000000E+00
	SALARY	12	DECIMAL	9	2 Y Y N -1.000000000000000E+00
	BONUS	13	DECIMAL	9	2 Y Y N -1.000000000000000E+00
	COMM	14	DECIMAL	9	2 Y Y N -1.000000000000000E+00
OTD76TB01_DEPT	ADMR3	ADMR3ADB	TD76TS01	5	-1.000000000000000E+00
0					
	Column Name	Col No	Col Type	Length	Scale Null Def FP
	DEPTNO	1	CHAR	3	0 N N N -1.000000000000000E+00
	DEPTNAME	2	VARCHAR	36	0 N N N -1.000000000000000E+00
	MGRNO	3	CHAR	6	0 Y Y N -1.000000000000000E+00
	ADMRDEPT	4	CHAR	3	0 N N N -1.000000000000000E+00
	LOCATION	5	CHAR	16	0 Y Y N -1.000000000000000E+00
OTD76TB02_EMP	ADMR3	ADMR3ADB	TD76TS02	14	-1.000000000000000E+00
0					
	Column Name	Col No	Col Type	Length	Scale Null Def FP
	EMPNO	1	CHAR	6	0 N N N -1.000000000000000E+00
	FIRSTNME	2	VARCHAR	12	0 N N N -1.000000000000000E+00
	MIDINIT	3	CHAR	1	0 N N N -1.000000000000000E+00
	LASTNAME	4	VARCHAR	15	0 N N N -1.000000000000000E+00
	WORKDEPT	5	CHAR	3	0 Y Y N -1.000000000000000E+00
	PHONENO	6	CHAR	4	0 Y Y N -1.000000000000000E+00
	HIREDATE	7	DATE	4	0 Y Y N -1.000000000000000E+00
	JOB	8	CHAR	8	0 Y Y N -1.000000000000000E+00
	EDLEVEL	9	SMALLINT	2	0 Y Y N -1.000000000000000E+00
	SEX	10	CHAR	1	0 Y Y N -1.000000000000000E+00
	BIRTHDATE	11	DATE	4	0 Y Y N -1.000000000000000E+00
	SALARY	12	DECIMAL	9	2 Y Y N -1.000000000000000E+00
	BONUS	13	DECIMAL	9	2 Y Y N -1.000000000000000E+00
	COMM	14	DECIMAL	9	2 Y Y N -1.000000000000000E+00
OTD76TB03_ACT	ADMR3	ADMR3ADB	TD76TS03	3	-1.000000000000000E+00
0					
	Column Name	Col No	Col Type	Length	Scale Null Def FP
	ACTNO	1	SMALLINT	2	0 N N N -1.000000000000000E+00

15697-L90 IBM DB2 Administration Tool for Z/OS Table Report PAGE 5

OName	Owner	T	DB Name	TS Name	Col's	Rows	Ch
0	Column Name	Col No	Col Type	Length	Scale	Null	Def FP
	ACTKWD	2	CHAR	6	0	N	N N -1.000000000000000E+00
	ACTDESC	3	VARCHAR	20	0	N	N N -1.000000000000000E+00

15697-L90 IBM DB2 Administration Tool for Z/OS

Index Report

PAGE 8

OIX Name	IX Owner	TB Name	TB Owner	U	Col's	CG	CD	C
TD76XA01	ADMR3	TD76TB01_DEPT	ADMR3	P	1	N	N	N
0	Column Name	Ordering						
	DEPTNO	A						
TD76XA02	ADMR3	TD76TB02_EMP	ADMR3	P	1	Y	Y	N
0	Column Name	Ordering						
	EMPNO	A						
TD76XA03	ADMR3	TD76TB03_ACT	ADMR3	P	1	N	N	N
0	Column Name	Ordering						
	ACTNO	A						
TD76XA04	ADMR3	TD76TB04_PROJ	ADMR3	P	1	N	N	N
0	Column Name	Ordering						
	PROJNO	A						
TD76XA05	ADMR3	TD76TB05_PROJACT	ADMR3	P	3	N	N	N
0	Column Name	Ordering						
	PROJNO	A						
	ACTNO	A						
	ACSTDATE	A						
TD76XA06	ADMR3	TD76TB06_EMPPROJACT	ADMR3	U	4	N	N	N
0	Column Name	Ordering						
	PROJNO	A						
	ACTNO	A						
	EMSTDATE	A						
	EMPNO	A						
TD76XA13	ADMR3	TD76TB13_PARTS	ADMR3	D	1	N	N	N
0	Column Name	Ordering						
	ITEMNUM	A						
TD76XA14	ADMR3	TD76TB14_DEPT	ADMR3	P	1	N	N	N
0	Column Name	Ordering						
	DEPTNO	A						
TD76XB01	ADMR3	TD76TB01_DEPT	ADMR3	D	1	N	N	N

11.2.3 STATUS

The STATUS command shows the activity that has been undertaken during the current session, and the details of the current subsystem to which you are currently connected. An example is shown in Figure 11-9.

```
DB2 Admin ----- DB0B DB2 Admin Status ----- 19:17
Option ==>

Current DB2 Admin status: Accessing the local system

Local DB2 subsystem name: DB0B
Userid                   : ADMR3
Current SQL ID           : ADMR3
Current SCHEMA           : ADMR3

DB2 release              : 1015
DB2 product              : DB2

Catalog qualifier        : SYSIBM - running directly on catalog tables
DDF location             : (blank) - running locally
Current server           : DB0B - local server
Remote subsystem name    : n/a

Execution totals          Counts
Prepare                  : 186
Describe                 : 201
Open                    : 24
Fetch                   : 274
Close                   : 24

Execute dynamically      : 162
- Set                   : 0
- Insert                : 133
- Update                : 0
- Delete                : 0

Execution totals          Counts
Prepare                  : 186
Describe                 : 201
Open                    : 24
Fetch                   : 274
Close                   : 24
Commit                  : 32
Rollback                 : 0
Connect                 : 1
Set                     : 4
User rows affected       : 133

Execute dynamically      : 162
- Set                   : 0
- Insert                : 133
- Update                : 0
- Delete                : 0
- Create                : 0
- Drop                  : 8
- Alter                 : 0
- Comment               : 0
- Label                 : 0
- Grant                 : 0
- Revoke                : 0
- Rename                : 0
- Commit                : 14
- Rollback              : 0
- Other dynamic         : 7

Use the RESET command to reset the counts
```

Figure 11-9 Session status

11.2.4 SEARCH

DB2 Administration Tool provides you with two methods of searching your results list. The first method uses the field underneath the column name, indicated by the asterisk. Figure 11-10, shows a search looking for all indexes starting with TD76XA from the list of indexes owned by ADMR3 (which was built using the ALL X command in Figure 11-6 on page 286)

```
DB2 Admin ----- DB0B Indexes ----- Row 1 from 20
Command ==> Scroll ==> PAGE

Commands: DIS STA STO ALL
Line commands:
T - Tables D - Database G - Storage group P - Plans C - Columns
DIS - Display index space STA - Start index space STO - Stop index space
? - Show all line commands

Select Index Name      Index      Table      Table      U      C C CC
                        Schema      Name      Schema      *      *  G D LM
                        *          *          *          *      * * * **
----->-----
TD76XA01      ADMR3      TD76TB01_DEPT      ADMR3      P      1 N N NN
TD76XA02      ADMR3      TD76TB02_EMP      ADMR3      P      1 Y Y NN
TD76XA03      ADMR3      TD76TB03_ACT      ADMR3      P      1 N N NN
TD76XA04      ADMR3      TD76TB04_PROJ      ADMR3      P      1 N N NN
TD76XA05      ADMR3      TD76TB05_PROJACT      ADMR3      P      3 N N NN
TD76XA06      ADMR3      TD76TB06_EMPPROJAC      ADMR3      U      4 N N NN
TD76XA13      ADMR3      TD76TB13_PARTS      ADMR3      D      1 N N NN
TD76XA14      ADMR3      TD76TB14_DEPT      ADMR3      P      1 N N NN
***** END OF DB2 DATA*****
```

Figure 11-10 Simple searching

The second method uses the SEARCH command. When you issue this command, a panel opens that allows you to enter more complex search criteria against all catalog fields for the object, and not just the columns that are present on the panel. For example, using the list of tables in Figure 11-5 on page 285, you could perform a search for all tables with a row length greater than 100 bytes. issue the SEARCH command, find the RECLENGTH column (second page), and enter the search conditions (Figure 11-11).

DB2 Admin ----- DB0B Search fields -----				Row 18 to 34 of 59
Command ==>				Scroll ==> PAGE
Select	Column Name	DB2 Col No	Srch Oper	Search Value
*		*	*	*

	PARENTS	18		*
	CHILDREN	19		*
	KEYCOLUMNS	20		*
	RECLENGTH	21	>	100
	STATUS	22		*
	KEYOBID	23		*
	LABEL	24		*
	CHECKFLAG	25		*
	CHECKRID	26		*
	AUDITING	27		*
	CREATEDBY	28		*
	LOCATION	29		*
	TBCREATOR	30		*
	TBNAME	31		*
	CREATEDTS	32		*
	ALTEREDTS	33		*
	DATA_CAPTURE	34		*

Figure 11-11 Entering SEARCH conditions

When you return to the table list, you are presented with the results of the search (Figure 11-12).

```
DB2 Admin ----- DBOB Tables, Views, and Aliases ----- Row 1 from 40
Command ==> Scroll ==> PAGE

Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols  Rows ChksC
    *                  *      *  *      *      *      *   **
-----
EMP          ADMR3    T ADMR3ADB TESTING    14      -1    2
TD76TB02_EMP ADMR3    T ADMR3ADB TD76TS02    14      -1    2
TD76TB12_EEMP ADMR3    T ADMR3ADB TD76TS12    16      -1    0
PLAN_TABLE   ADMR3    T DSN00148 PLANRTAB    64      -1    0
***** END OF DB2 DATA*****
```

Figure 11-12 Search results

To return to the original list, issue SEARCH RESET and the original list is displayed.

11.2.5 DB2

DB2 is a simple primary command that allows you to run a DB2 command from any panel by prefixing this command to the DB2 command. In our example, we prefix DBS to -DIS THREAD(*) (Figure 11-13).

```
DB2 Admin ----- DBOB Tables, Views, and Aliases ----- Row 1 from 40
Command ==> DB2 -DIS THREAD(*)                               Scroll ==> PAGE

Commands: GRANT MIG ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel   Name                Schema   T DB Name  TS Name    Cols      Rows ChksC
   *                *      *  *      *      *      *      **
-----
      EMP                ADMR3    T ADMR3ADB TESTING    14        -1     2
      TD76TB02_EMP        ADMR3    T ADMR3ADB TD76TS02    14        -1     2
      TD76TB12_EEMP        ADMR3    T ADMR3ADB TD76TS12    16         -1     0
      PLAN_TABLE          ADMR3    T DSN00148 PLANRTAB   64         -1     0
***** END OF DB2 DATA*****
```

Figure 11-13 DB2 command

The DB2 command is executed and you are presented with the results (Figure 11-14).

```

DB2 Admin ----- DB0B Browse DB2 Command Output --- Line 00000000 Col 001080
Command ==> Scroll ==> PAGE

-DIS THREAD(*)

***** Top of Data*****
DSNV401I  -DB0B DISPLAY THREAD REPORT FOLLOWS -
DSNV402I  -DB0B ACTIVE THREADS -
NAME      ST A  REQ ID          AUTHID    PLAN      ASID  TOKEN
DISCONN   DA *  1794 NONE          NONE      DISTSERV  0089   416
V471-USIBMSC.SCPDB0B.C6C3992B7D3E=416
RRSAF     T    6645 DB0BADMT_DMN STC      ?RRSAF    008A    2
RRSAF     T      5 DB0BADMT_II. STC      ?RRSAF    008A    3
TSO       N     41 DB2R8          DB2R8      0093    0
TSO       T    112 DB2R8          DB2R8      DSNEPCS   0093   358
TSO       T *   105 ADMR3          ADMR3      ADB       0094   414
TSO       N     195 XMLR5          XMLR5      0091    0
TSO       N     154 XMLR2          XMLR2      008E    0
DISPLAY ACTIVE REPORT COMPLETE
DSN9022I  -DB0B DSNVDT '-DIS THREAD' NORMAL COMPLETION
***** Bottom of Data *****

```

Figure 11-14 DB2 command result

Upon exiting this panel, you are taken back to the panel from which you entered the command.

Tip: When you enter a DB2 Administration Tool primary command that has the same name as an ISPF command, the ISPF command is executed first. To bypass the TSO command processor, enter the primary command with a prefix of the greater than symbol (>), which is a TSO escape character.

11.3 Features you may have missed

In this section, we look at powerful but small features of the product that can bring extra value to an organization.

11.3.1 Using the DB2 Administration Tool panels for SQL

Most DB2 installations have a set of SQL that they occasionally run to identify DB2 objects that may require some maintenance, for example, overallocated and underallocated data sets, badly organized table spaces, and so on. This SQL has evolved over time and is of value to you in identifying any areas of concern. These queries are run on an intermittent basis. DB2 Administration Tool can add extra value to these queries by using its capabilities to display the results and to give you the ability to start remedying the situation.

In the DB2 Administration Tool, you can use option 2 to run SQL (Figure 11-15).

```
ADB22 min ----- Execute SQL Statements ----- 20:00
Option ==>

    1 - Execute SQL statements from screen input          DB2 System: DB0B
    2 - Run or Explain SQL statements                    DB2 SQL ID: ADMR3
    3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
    4 - Create/drop/label/comment on objects
    5 - Grant/revoke privileges on objects
```

Figure 11-15 Execute SQL Statements panel

When you select option 1, the panel shown in Figure 11-16 opens. From this panel, you can enter user written SQL to query the catalog to run queries.

```
DB2 Admin ----- Execute SQL Statements from Screen Input -----
12:52
Command ==>

SQL statement:

DB2 System: DB0B
DB2 SQL ID: ADMR3

Press ENTER to execute the SQL statement, or enter EDIT on the command line to
edit it. Use command CAPS (OFF/ON) to change the caps mode.
```

Figure 11-16 SQL entry panel

This panel is a free-form entry screen for SQL, but with the additional benefit that you can present the results using DB2 Administration Tool panels using a “hidden” field on the panel.

For example, assume you want to find all the indexes for a certain database where the clusterratio has fallen below 95%. With this list, you want to build REORG utility jobs, which obviously have to be at the table space level to increase the clusterratio of the index. On the panel, enter your SQL and then add the DB2 Administration panel that corresponds to the data, for example, ADB21S, which relates to table spaces and ADB21T tables. The panel is shown in Figure 11-17.

DB2 Admin ----- Execute SQL Statements from Screen Input ----- 14:17	
Command ==>	
ADB21S	DB2 System: DB0B
SQL statement:	DB2 SQL ID: ADMR3
SELECT * FROM SYSIBM.SYSTABLESPACE	
WHERE DBNAME LIKE 'ADMR3%'	
AND NAME IN (SELECT TSNAME FROM SYSIBM.SYSTABLES	
WHERE NAME IN (SELECT TBNAME FROM SYSIBM.SYSINDEXES	
WHERE CLUSTERRATIOF < 95))	
>	
Press ENTER to execute the SQL statement, or enter EDIT on the command line to edit it. Use command CAPS (OFF/ON) to change the caps mode.	

Figure 11-17 Using the hidden field

Tip: Enter EDIT on the Command line to open an ISPF Edit session.

Tip: If you specify the incorrect panelid, the DB2 Administration Tool tries to match the columns to the appropriate fields but the information is in the wrong columns. For example, the table space name is displayed in the table name in the ADB21T panel because that field relates to the NAME column.

This action causes the resultant data to be displayed on the table space panel for the tool (Figure 11-18).

```

DB2 Admin ----- DBOB Table Spaces ----- Row 1 to 9 of 9
Command ==> UTIL Scroll ==> PAGE

Commands: GRANT MIG DIS STA STO ALL
Line commands:
T - Tables D - Database A - Auth G - Storage group ICS - Image copy status
DIS - Display table space STA - Start table space STO - Stop table space
? - Show all line commands

Select Name DB Name Parts Bpool L E S I C Tables Act. pages Segsz T L
* * * * * * * * * * * * * * * *
-----
TD76TS01 ADMR3ADB 0 BPO P N A N N 1 -1 4 Y
TD76TS02 ADMR3ADB 5 BPO P N A N N 1 -1 0 Y
TD76TS03 ADMR3ADB 0 BPO R N A N N 1 -1 4 Y
TD76TS04 ADMR3ADB 0 BPO R N A N N 1 -1 4 Y
TD76TS05 ADMR3ADB 0 BPO R N A N N 1 -1 4 Y
TD76TS06 ADMR3ADB 0 BPO R N A N N 1 -1 4 Y
TD76TS13 ADMR3ADB 0 BPO P N A N N 1 -1 4 Y
TD76TS14 ADMR3ADB 2 BPO P N A N N 1 -1 4 G Y
TESTING ADMR3ADB 5 BPO P N A N N 1 -1 0 Y
***** END OF DB2 DATA *****

```

Figure 11-18 SQL results in DB2 Administration Tool panel

In Figure 11-18 on page 297, you can use the UTIL command to generate utilities for the objects listed, which takes you to the table space utilities panel (Figure 11-19), where you can generate your Reorgs.

```

DB2 Admin ----- DB0B Table Space Utilities ----- 14:35
Option ==>
Top of data
Execute utility on
    all the selected table spaces
                                DB2 System: DB0B
                                DB2 SQL ID: ADMR3
                                More:      +
    C - Copy full                CI - Copy incremental    C2 - Copytocopy
    CC - Copy concurrent
    E - Mergecopy                EN - Mergecopy newcopy
    K - Check index              KD - Check data          KL - Check LOB
    M - Modify                    NW - Repair Auxwarn      NX - Repair Auxcheckpend
    N - Repair nocopypend        NA - Repair nocheckpend  NB - Repair norcvrpend
                                NR - Repair noreorgpend
    O - Reorg                    OU - Reorg unload only
    OC - Reorg w/Inline Copy
    P - Report recovery          Q - Quiesce

Utility control options:
Review/change options           : NO    (Yes/No)
Generate work statement list    : NO    (Yes/No)
Generate template statements    : NO    (Yes/No)
Generate modify after copy      : NO    (Yes/No)

```

Figure 11-19 Table space utility generation

11.3.2 Querying the catalog using nonstandard criteria

When querying the DB2 catalog, the majority of accesses to the catalog are made by using the standard SQL, such as owner of objects, database names, and so on. These options are covered by the standard selection criteria fields. Occasionally, there is a requirement to access the catalog using different criteria. An example of this situation is if an index runs out of space, you have the data set name for the index, and therefore the index space name. Using the X option on the System Catalog panel and the index space name may not result in any rows being returned.


```

DB2 Admin ----- DB0B System Catalog ----- 16:27
Option ==>

Object options:
  AO - Authorization options
  G - Storage groups
  D - Databases
  S - Table spaces
  T - Tables, views, and aliases
  V - Views
  A - Aliases
  Y - Synonyms
  X - Indexes
  C - Columns
  N - Constraints
  DS - Database structures
  PDC - DB2 pending definition changes

  P - Plans
  L - Collections
  K - Packages
  M - DBRMs
  H - Schemas
  E - User defined data types
  F - Functions
  O - Stored procedures
  J - Triggers
  Q - Sequences
  DSP - DS with plans and packages

More: +
DB2 System: DB0B
DB2 SQL ID: ADMR3

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name ==> SYSTABLESPACESTAT% > Grantor ==> >
Owner ==> > Grantee ==> >
In D/L/H ==> >
And/or other selection criteria (option xC shows you columns for option x)
Column ==> > Operator ==> Value ==>

```

An example of the data set names are shown in Example 11-2.

[illegible]

Using one of the index space names, you can find the index to which the name relates (Figure 11-21). Enter the column of the catalog table that you want to query and then enter the value to use.

```
DB2 Admin ----- DBOB System Catalog ----- 17:06
Option ==> X

More: -

G - Storage groups          P - Plans
D - Databases              L - Collections
S - Table spaces           K - Packages
T - Tables, views, and aliases M - DBRMs
V - Views                  H - Schemas
A - Aliases                E - User defined data types
Y - Synonyms               F - Functions
X - Indexes                O - Stored procedures
C - Columns                J - Triggers
N - Constraints            Q - Sequences
DS - Database structures   DSP - DS with plans and packages
PDC - DB2 pending definition changes

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name      ==>          > Grantor ==>          >
Owner     ==> ADMR3%   > Grantee ==>          >
In D/L/H ==>          >

And/or other selection criteria (option xC shows you columns for option x)
Column    ==> INDEXSPACE > Operator ==> =      Value ==> TD761KS6
```

Figure 11-21 Finding the index

The result of this query is shown in Figure 11-22.

```
ADB21X in ----- DBOB Indexes ----- Row 1 to 1 of 1
Command ==>                               Scroll ==> PAGE

Commands: DIS STA STO ALL
Line commands:
T - Tables D - Database G - Storage group P - Plans C - Columns
DIS - Display index space STA - Start index space STO - Stop index space
? - Show all line commands

Select Index Name          Index          Table          C C C C
      *              Schema Table Name      Schema  U  Cols G D L M
----->-----
      XD76TB04_LONG_INDE ADMR3      TD76TB04_PROJ      ADMR3  U    5 N N Y N
***** END OF DB2 DATA *****
```

Figure 11-22 Result of using other selection criteria to find an index

If you cannot remember the column to use, enter the suffix C to your command and you are presented with a list of columns valid for that catalog table. An example (using the TC command) is shown in Figure 11-23.

```
ADB21DC n -- DB0B Columns in TABLE: SYSIBM.SYSTABLE > ----- Row 1 to 15 of 59
Command ==>                                           Scroll ==> PAGE
```

Line commands:

T - Tables X - Indexes A - Auth GR - Grant H - Homonyms

Select	Column Name	Col No	Col Type	Length	Scale	Null
*		* *			*	* *
-----	-----	-----	-----	-----	-----	-----
	NAME	1	VARCHAR	128	0	N
	CREATOR	2	VARCHAR	128	0	N
	TYPE	3	CHAR	1	0	N
	DBNAME	4	VARCHAR	24	0	N
	TSNAME	5	VARCHAR	24	0	N
	DBID	6	SMALLINT	2	0	N
	OBID	7	SMALLINT	2	0	N
	COLCOUNT	8	SMALLINT	2	0	N
	EDPROC	9	VARCHAR	24	0	N
	VALPROC	10	VARCHAR	24	0	N
	CLUSTERTYPE	11	CHAR	1	0	N
	CLUSTERRID	12	INTEGER	4	0	N
	CARD	13	INTEGER	4	0	N
	NPAGES	14	INTEGER	4	0	N
	PCTPAGES	15	SMALLINT	2	0	N

Figure 11-23 Which column name to use



Part 6

DB2 10 for z/OS support

In this part, we discuss and provide examples of DB2 Administration Tool support for new functions introduced by DB2 10 for z/OS.

This part contains the following chapters:

- ▶ Chapter 12, “Temporal tables” on page 305
- ▶ Chapter 13, “Security” on page 357
- ▶ Chapter 14, “Physical design” on page 417
- ▶ Chapter 16, “General options” on page 483



Temporal tables

DB2 10 for z/OS introduces support for temporal tables and data versioning. In this chapter, we describe what a temporal table is and how you can create a bi-temporal table and alter an existing table to become a temporal table using the DB2 Administration Tool. We walk through an example to show the before and after of the table modification and the results of the temporal table data versioning to your application data.

This chapter includes the following topics:

- ▶ Temporal tables and versioning
- ▶ DB2 Administration Tool support of the temporal table
- ▶ Using the CT function to create a bi-temporal table
- ▶ Using the AL line command to convert a table to a temporal table

12.1 Temporal tables and versioning

Starting with the DB2 10 for z/OS new-function mode, you can define temporal attributes for a table and enable versioning of the table's data. A temporal table is a base table that is defined with a period or periods, and in some cases, versioning. A period is an interval of time that is defined by two columns of a table. The first column is the beginning time value of a period, and the second column is the ending time value of a period.

DB2 10 for z/OS defines a notion of a period, which is a time interval represented by a start time and an end time. DB2 10 for z/OS new function mode supports two types of periods, which are the system time (SYSTEM_TIME) period and the business time (BUSINESS_TIME) period.

The SYSTEM_TIME period is a system-maintained period in which DB2 maintains the beginning and ending timestamp values for a row. The begin time column contains the timestamp value for when a row is created. The end time column contains the timestamp value for when a row is updated or deleted.

The BUSINESS_TIME period also is called an application period. You, instead of DB2, maintain the beginning and ending values for a row. The begin time column contains the value for when a row is valid from. The end time column contains the value for when a row stops being valid.

The system period temporal table has the SYSTEM_TIME period defined on the table. The business period temporal table has the BUSINESS_TIME period defined on the table. It is a bi-temporal table when the table has both SYSTEM_TIME and BUSINESS_TIME period, an associated history table, and versioning defined.

The SYSTEM_TIME period is meaningful because of versioning. Versioning specifies that the old rows are archived into another table. The table that contains the current active rows of a table is called the system period temporal table. The table that contains the archived rows is called the history table. You must create a history table when you define versioning on an existing table.

12.2 DB2 Administration Tool support of the temporal table

You can use the CT (create Table) function to create a table with a SYSTEM TIME period or BUSINESS TIME period, and use ALT or the AL line command to change a non-temporal table to a temporal period table. In this book, we show both the CT function and the AL line command's support for temporal table.

12.3 Using the CT function to create a bi-temporal table

A bi-temporal table is a base table that contains both the defined SYSTEM_TIME period and BUSINESS_TIME period. In the following example, we show you the steps to create a bi-temporal table using the Administration Tool CT (Create Table) function.

Using option 2.4 from the DB2 Administration main menu (ADB2) takes you to the Create/Drop/Label/Comment On Objects panel (ADB26). The CT (create table) function on the Create/Drop/Label/Comment On Objects panel has been enhanced to allow you to create temporal table (Figure 12-1).

```

ADB26 min ----- VA1A Create/Drop/Label/Comment On Objects -----
Option ==> CT

                                     More:      +
                                     DB2 System: VA1A
                                     DB2 SQL ID: SYSADM

CREATE                                DROP
CG - Storage group                   DG - Storage group
CD - Database                        DD - Database
CS - Table space                     DS - Table space
CT - Table                           DT - Table
CV - View                            DV - View
CL - Alias                           DL - Alias
CX - Index                           DX - Index
CY - Synonym                         DY - Synonym
CA - Auxiliary table
CE - Distinct type
CJ - Trigger
CF - Function
CO - Stored procedure
CM - Materialized table
CQ - Sequence
CTR - Trusted context
CRO - Role
LABEL                                COMMENT (remark)

```

Figure 12-1 Create/Drop/Label/Comment On Objects panel

When you press Enter, the Create Table panel (ADB26CT) opens, which allows you to specify the schema, name of the table you are creating, and all the column specifications. In our example, we create a bi-temporal table that has nine columns, including the beginning and ending columns for SYSTEM_TIME and BUSINESS_TIME (Figure 12-2).

```

ADB26CT n ----- VA1A Create Table -----
Command ==>

CREATE TABLE

Schema . . . . . ADMR2      >          (default is ADMR2)
Name   . . . . . CUST_COVERAGE  >  (? to look up)

LIKE
Schema . . . . .           >
Name   . . . . .           >  (? to look up)
Identity attrs .          (Include identity attributes, Yes/No)
Row chg attrs . .        (Include row change TS attributes,
Yes/No)
As model . . . . .        (Use as model only, Yes/No)

(
Number of columns . . . 9

```

Figure 12-2 Create CUST_COVERAGE table

When you press Enter, the Create Table Columns panel (ADB26CTF) opens, where you can specify the column name, data type, length of the column, whether the column can contains null value, the default value where it is applicable, and the operation type. In our example, there are nine columns in the table and we need to enter the information for each column that is inserted (Figure 12-3).

```

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==>                                         Scroll ==> CSR

Schema . . ADMR2      >          Database . . .
Name . . . CUST_COVERAGE      > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands: M - Move A - After B - Before
               Inn - Insert U - Update D - Delete Rnn - Repeat

Select Column Name      Col Type      Length Scale Null D Col No Type
          *              *              *      * *      *
-----> -----> -----> -----> -----> -----> ----->
          ?              ?              0      0 ?      ?      1 INSERT
          ?              ?              0      0 ?      ?      2 INSERT
          ?              ?              0      0 ?      ?      3 INSERT
          ?              ?              0      0 ?      ?      4 INSERT
          ?              ?              0      0 ?      ?      5 INSERT
          ?              ?              0      0 ?      ?      6 INSERT
          ?              ?              0      0 ?      ?      7 INSERT
          ?              ?              0      0 ?      ?      8 INSERT
          ?              ?              0      0 ?      ?      9 INSERT
***** END OF DB2 DATA *****

```

Figure 12-3 Specifying table columns

We enter the column name and then enter the column data type and its data length for the column. You can specify the scale for column, and its data type is DECIMAL, DECFLOAT, or TIMESTAMP.

The valid default values that you can specify on the panel are as follows:

- ▶ 1- String constant
- ▶ 2- Floating point constant
- ▶ 3- Decimal constant
- ▶ 4- Integer constant
- ▶ 5- hexadecimal string
- ▶ 6- UX string
- ▶ N- None
- ▶ Y- Yes
- ▶ B- Yes
- ▶ S- SQLID
- ▶ U- USER
- ▶ A:D - rowid
- ▶ I:J Identity Column
- ▶ L- SECLABEL
- ▶ E- Generated Always for each row on update as row change timestamp
- ▶ F- Generated By default for each row on update as row change timestamp

- ▶ Q- GENERATED ALWAYS AS ROW BEGIN
- ▶ R- GENERATED ALWAYS AS ROW END
- ▶ X- GENERATED ALWAYS AS TRANSACTION START ID

Q, R, and X are the new values for the temporal table support.

In our example, we create a bi-temporal table that contains the following columns with the SYSTEM_TIME period and BUSINESS_TIME period:

```
CUSTO_ID - INTEGER NOT NULL WITH NO DEFAULT
CATEGORY - CHAR (2) NOT NULL WITH NO DEFAULT
DEDUCTABLE - DECIMAL(15,2) NOT NULL WITH NO DEFAULT
COVERAGE - DECIMAL(15,2) NOT NULL WITH NO DEFAULT
```

To create a SYSTEM_TIME temporal table, you need to define the following three columns:

- ▶ Beginning SYSTEM time column
This is a column that is maintained by DB2. The column is defined as GENERATED ALWAYS AS ROW BEGIN. The data type of the column needs to be TIMESTAMP(12) NOT NULL WITHOUT TIMEZONE.
- ▶ Ending SYSTEM time column;
This a column that is maintained by DB2. The column is defined as GENERATED ALWAYS AS ROW END. The data type of the column needs to be TIMESTAMP(12) NOT NULL WITHOUT TIMEZONE.
- ▶ Transaction Start ID column;
This a column that is maintained by DB2. The column is defined as GENERATED ALWAYS AS TRANSACTION START ID. The data type of the column needs to be TIMESTAMP(12) WITHOUT TIMEZONE.

To create a BUSINESS_TIME temporal table, you need to define the following two columns:

- ▶ Beginning BUSINESS time column
This is a column that is maintained by you. The column data type of the column can be either TIMESTAMP or DATE.
- ▶ Ending BUSINESS time column
This is a column that is maintained by you. The data type of the column can be either TIMESTAMP or DATE.

On the Create Table Columns panel (ADB26CTF), we enter the column specification for the CUSTID, CATEGORY, DEDUCTABLE, COVERAGE, and SYS_STA columns. The SYS_STA column is specified as TIMESTAMP(12) NOT NULL and the default is Q for the GENERATED ALWAYS AS ROW BEGIN (Figure 12-4).

Note: The SYS_STA column data type is TIMESTAMP(12). The 12 is specified under Scale rather than the Length. For the TIMESTAMP column, Scale is the number of fractional second digits

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==> Scroll ==> CSR

Schema . . ADMR2 > Database . . .
Name . . . CUST_COVERAGE > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands: M - Move A - After B - Before
Inn - Insert U - Update D - Delete Rnn - Repeat

Select	Column Name	Col Type	Length	Scale	Null	D	Col	No	Operation
*	*	*	*	*	*	*	*	*	*
----->	----->	----->	----->	----->	----->	----->	----->	----->	----->
	CUST_ID	INTEGER	0	0	N	N	1		INSERT
	CATEGORY	CHAR	2	0	N	N	2		INSERT
	DEDUCTABLE	DECIMAL 15		2	N	N	3		INSERT
	COVERAGE	DECIMAL 15		2	N	N	4		INSERT
	SYS_STA	TIMESTAMP	0	12	N	Q	5		INSERT
	?	?	0	0	?	?	6		INSERT
	?	?	0	0	?	?	7		INSERT
	?	?	0	0	?	?	8		INSERT
	?	?	0	0	?	?	9		INSERT
***** END OF DB2 DATA *****									

Figure 12-4 Enter the first four columns and the SYS_STA column for CUST_COVERAGE

Press Enter after the first five columns are specified. Notice that the Col Type of the SYS_STA column is changed from TIMESTAMP to TIMESTMP and length of the SYS_STA column is 13 (Figure 12-5).

Note: The column type TIMESTAMP and TIMESTMP are both valid data type and are interchangeable. For the length of the timestamp column, DB2 Administration Tool calculates the length value based on the Scale value. Any value entered on the length field is ignored.

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==> Scroll ==> CSR

Schema . . ADMR2 > Database . . .
Name . . . CUST_COVERAGE > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands: M - Move A - After B - Before
Inn - Insert U - Update D - Delete Rnn - Repeat

Select	Column Name	Col Type	Length	Scale	Null	D	Col	No	Operation
*	*	*	*	*	*	*	*	*	*
----->	----->	----->	----->	----->	----->	----->	----->	----->	----->
*	CUST_ID	INTEGER	4	0	N	N	1	INSERT	
*	CATEGORY	CHAR	2	0	N	N	2	INSERT	
*	DEDUCTABLE	DECIMAL	15	2	N	N	3	INSERT	
*	COVERAGE	DECIMAL	15	2	N	N	4	INSERT	
*	SYS_STA	TIMESTMP	13	12	N	Q	5	INSERT	
	?	?	0	0	?	?	6	INSERT	
	?	?	0	0	?	?	7	INSERT	
	?	?	0	0	?	?	8	INSERT	
	?	?	0	0	?	?	9	INSERT	
***** END OF DB2 DATA *****									

Figure 12-5 SYS_STA length is changed from 0 to 13

To define the SYS_STA column to be WITHOUT TIMEZONE, we issue the U (Update) line command next to the SYS_STA (Figure 12-6).

```
ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==>                                         Scroll ==> CSR

Schema . . ADMR2      >          Database . . .
Name . . . CUST_COVERAGE      > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands: M - Move A - After B - Before
Inn - Insert U - Update D - Delete Rnn - Repeat

Select Column Name      Col Type      Length Scale Null D Col No Type      Operation
*                      *              *      * *      *      * *
-----> -----

*      CUST_ID          INTEGER          4      0 N      N      1 INSERT
*      CATEGORY         CHAR              2      0 N      N      2 INSERT
*      DEDUCTABLE        DECIMAL          15      2 N      N      3 INSERT
*      COVERAGE         DECIMAL          15      2 N      N      4 INSERT
U     SYS_STA          TIMESTMP          13     12 N      Q      5 INSERT
      ?                ?                  0      0 ?      ?      6 INSERT
      ?                ?                  0      0 ?      ?      7 INSERT
      ?                ?                  0      0 ?      ?      8 INSERT
      ?                ?                  0      0 ?      ?      9 INSERT

***** END OF DB2 DATA *****
```

Figure 12-6 Issue the U line command next to the SYS_STA column

When you press Enter, the Create Table Column Number panel (ADB26CUU) opens. In our case, it is Create Table Column Number 5, which is used to update the fifth column of the table. Enter NO for the WITH TIME ZONE clause, meaning that the SYS_STA column is defined as WITHOUT TIMEZONE (Figure 12-7). Press Enter.

ADB26CUU ----- VA1A Create Table Column Number 5 -----	
Command ==>	
More: +	
CREATE TABLE	Schema . . . ADMR2 >
	Name CUST_COVERAGE >
Column name . . . SYS_STA	> (Column number 5)
Data type Timestmp	(Built-in only)
Data length . . .	(Built-in only)
Precision	(FLOAT and DECIMAL only)
Scale 12	(DECIMAL and timestamp types only)
Type schema . . .	> (User-defined type schema)
Type name	> (? to look up)
WITH TIME ZONE . NO	(Yes/No - for Timestmp only)
Allow Nulls . . . NO	(Yes or blank--nullable, No-NOT NULL)
FOR ? DATA . . .	(B-Bit, S-SBCS, M-Mixed, blank-N/A)
WITH DEFAULT . .	(Yes, No, L (SECLABEL) or enter value below)
Default value . .	
GENERATED Q	(A-ALWAYS, D-DEFAULT, I-ALWAYS AS IDENTITY, J-DEFAULT AS IDENTITY, E-ALWAYS AS UPD Timestmp, F-DEFAULT AS UPD Timestmp, Q-ALWAYS AS ROW BEGIN, R-ALWAYS AS ROW END, X-ALWAYS AS TRANSACTION START ID)
FIELDPROC	
Program name . .	(Optional)
Program parm . .	>
Hidden	(Yes/No)

Figure 12-7 Enter NO for WITH TIME ZONE

When you press Enter, you return to the Create Table Columns panel (ADB26CTF). The Operation type of the SYS_STA column is changed from INSERT to UPDATE (Figure 12-8).

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==> Scroll ==> CSR

Schema . . ADMR2 > Database . . .
Name . . . CUST_COVERAGE > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands: M - Move A - After B - Before
Inn - Insert U - Update D - Delete Rnn - Repeat

Select	Column Name	Col Type	Length	Scale	Null	D	Col	No	Operation
*	*	*	*	*	*	*	*	*	*
----->-----									
*	CUST_ID	INTEGER	4	0	N	N		1	INSERT
*	CATEGORY	CHAR	2	0	N	N		2	INSERT
*	DEDUCTABLE	DECIMAL	15	2	N	N		3	INSERT
*	COVERAGE	DECIMAL	15	2	N	N		4	INSERT
*	SYS_STA	TIMESTAMP	13	12	N	Q		5	UPDATE
	?	?	0	0	?	?		6	INSERT
	?	?	0	0	?	?		7	INSERT
	?	?	0	0	?	?		8	INSERT
	?	?	0	0	?	?		9	INSERT
***** END OF DB2 DATA *****									

Figure 12-8 Operation type of SYS_STA changed to UPDATE

Continue to specify the SYS_END column. The SYS_END column is defined as data type TIMESTAMP(12) with the default R - GENERATED ALWAYS AS ROW END. Because you want to update the SYS_END column as WITHOUT TIMEZONE, enter the U (Update) line command after you enter the column name, Col Type, Scale, NOT NULL, and default R (Figure 12-9).

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9

Command ==> Scroll ==> CSR

Schema . . ADMR2 > Database . . .

Name . . . CUST_COVERAGE > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART

Line commands: M - Move A - After B - Before

Inn - Insert U - Update D - Delete Rnn - Repeat

Select		Column Name	Col Type	Length	Scale	Null	D	Col	No	Operation
		*	*	*	*	*	*		*	*
----->-----										
*		CUST_ID	INTEGER	4	0	N		N	1	INSERT
*		CATEGORY	CHAR	2	0	N		N	2	INSERT
*		DEDUCTABLE	DECIMAL	15	2	N		N	3	INSERT
*		COVERAGE	DECIMAL	15	2	N		N	4	INSERT
*		SYS_STA	TIMESTAMP	13	12	N		Q	5	UPDATE
U		SYS_END	TIMESTAMP	0	12	N		R	6	INSERT
		?	?	0	0	?		?	7	INSERT
		?	?	0	0	?		?	8	INSERT
		?	?	0	0	?		?	9	INSERT

***** END OF DB2 DATA *****

Figure 12-9 Specify SYS_END column

```

ADB26CUU ----- VA1A Create Table Column Number      6 -----
Command ==>

                                     More:    +
CREATE TABLE                               Schema . . . ADMR2    >
                                           Name . . . CUST_COVERAGE    >
Column name . . . SYS_END                > (Column number      6)

Data type . . . TIMESTMP                 (Built-in only)
Data length . . .                       (Built-in only)
Precision . . .                         (FLOAT and DECIMAL only)
Scale . . . . . 12                      (DECIMAL and timestamp types only)
Type schema . . .                       > (User-defined type schema)
Type name . . .                         > (? to look up)
WITH TIME ZONE . NO                   (Yes/No - for TIMESTAMP only)

Allow Nulls . . . NO   (Yes or blank--nullable, No-NOT NULL)
FOR ? DATA . . .     (B-Bit, S-SBCS, M-Mixed, blank-N/A)
WITH DEFAULT . . .     (Yes, No, L (SECLABEL) or enter value below)
Default value . . .
GENERATED . . . R (A-ALWAYS,           D-DEFAULT,
                  I-ALWAYS AS IDENTITY, J-DEFAULT AS IDENTITY,
                  E-ALWAYS AS UPD TIMESTAMP, F-DEFAULT AS UPD TIMESTAMP,
                  Q-ALWAYS AS ROW BEGIN,   R-ALWAYS AS ROW END,
                  X-ALWAYS AS TRANSACTION START ID)

FIELDPROC
Program name . . .         (Optional)
Program parm . . .
Hidden . . . . .          (Yes/No)

```

Chapter 12. Temporal tables **317**

```

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==> Scroll ==> CSR

Schema . . ADMR2      > Database . . .
Name . . . CUST_COVERAGE      > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands:  M - Move  A - After  B - Before
                Inn - Insert  U - Update  D - Delete  Rnn - Repeat

Select Column Name      Col Type      Length Scale  Null D Col No Type
      *                  *            *      *      *      *      *
-----> -----
*      CUST_ID           INTEGER        4          0 N      N      1 INSERT
*      CATEGORY          CHAR            2          0 N      N      2 INSERT
*      DEDUCTABLE        DECIMAL         15          2 N      N      3 INSERT
*      COVERAGE          DECIMAL         15          2 N      N      4 INSERT
*      SYS_STA           TIMESTMP        13         12 N      Q      5 UPDATE
*      SYS_END           TIMESTMP        13         12 N      R      6 UPDATE
*      ?                 ?              0          0 ?      ?      7 INSERT
*      ?                 ?              0          0 ?      ?      8 INSERT
*      ?                 ?              0          0 ?      ?      9 INSERT

***** END OF DB2 DATA *****

```

To continue defining the remaining columns of the CUST_COVERAGE table, we specify the BUS_STA and BUS_END column with data type DATE with NOT NULL WITHOUT DEFAULT for the BUSINESS_TIME period and the CREATE_ID as the transaction start ID for the SYSTEM_TIME period.

We define the CREATE_ID column with data type TIMESTAMP(12) NOT NULL with DEFAULT X - GENERATED ALWAYS AS TRANSACTION ID. Because the CREATE_ID needs to be defined as WITHOUT TIMEZONE, we enter the U(pdate) line command next to the CREATE_ID column so that we can modify the definition of CREATE_ID column (Figure 12-12).

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==> Scroll ==> CSR

Schema . . ADMR2 > Database . . .
Name . . . CUST_COVERAGE > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands: M - Move A - After B - Before
Inn - Insert U - Update D - Delete Rnn - Repeat

Select		Column Name	Col Type	Length	Scale	Null	D	Col	No	Type	Operation
		*	*	*	*	*	*	*	*	*	*
----->-----											
*		CUST_ID	INTEGER	4	0	N		N		1	INSERT
*		CATEGORY	CHAR	2	0	N		N		2	INSERT
*		DEDUCTABLE	DECIMAL	15	2	N		N		3	INSERT
*		COVERAGE	DECIMAL	15	2	N		N		4	INSERT
*		SYS_STA	TIMESTMP	13	12	N		Q		5	UPDATE
*		SYS_END	TIMESTMP	13	12	N		R		6	UPDATE
		BUS_STA	DATE	0	0	N		N		7	INSERT
		BUS_END	DATE	0	0	N		N		8	INSERT
U		CREATE_ID	TIMESTMP	0	12	N		X		9	INSERT
***** END OF DB2 DATA *****											

Figure 12-12 Specifying BUS_STA, BUS_END, and the CREATE_ID column

After pressing Enter, the Create Table Column Number panel (ADB26CUU) opens, where you update the ninth column of the table. Enter NO for the WITH TIME ZONE clause, meaning that the CREATE_ID column is defined as WITHOUT TIMEZONE (Figure 12-13).

```

ADB26CUU ----- VA1A Create Table Column Number      9 -----
17:11
Command ==>

CREATE TABLE                                     Schema . . . ADMR2      >
                                                    Name . . . . CUST_COVERAGE      >
Column name . . . CREATE_ID                    > (Column number      9)

Data type . . . . TIMESTMP                        (Built-in only)
Data length . . . .                               (Built-in only)
Precision . . . .                               (FLOAT and DECIMAL only)
Scale . . . . . 12                             (DECIMAL and timestamp types only)
Type schema . . .                               > (User-defined type schema)
Type name . . . .                               > (? to look up)
WITH TIME ZONE . NO                             (Yes/No - for TIMESTAMP only)

Allow Nulls . . . NO (Yes or blank--nullable, No-NOT NULL)
FOR ? DATA . . . (B-Bit, S-SBCS, M-Mixed, blank-N/A)
WITH DEFAULT . . (Yes, No, L (SECLABEL) or enter value below)
Default value . .
GENERATED . . . . X (A-ALWAYS,                      D-DEFAULT,
                    I-ALWAYS AS IDENTITY,          J-DEFAULT AS IDENTITY,
                    E-ALWAYS AS UPD TIMESTAMP,      F-DEFAULT AS UPD TIMESTAMP,
                    Q-ALWAYS AS ROW BEGIN,          R-ALWAYS AS ROW END,
                    X-ALWAYS AS TRANSACTION START ID)

FIELDPROC
Program name . . (Optional)
Program parm . .
                                                    >

Hidden . . . . . (Yes/No)

```

Figure 12-13 Update CREATE_ID to WITHOUT TIMEZONE

After pressing Enter, you return to the Create Table Columns panel (ADB26CTF). The Scale of the CREATE_ID column is changed to 13 and the Operation type of the CREATE_ID column is changed from INSERT to UPDATE (Figure 12-14).

```

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==>                                         Scroll ==> CSR

Schema . . ADMR2      >          Database . . .
Name . . . CUST_COVERAGE      > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands:  M - Move  A - After  B - Before
                Inn - Insert  U - Update  D - Delete  Rnn - Repeat

                                                    Operation
Select Column Name      Col Type      Length Scale  Null D Col No Type
      *                  *            *      * *      *      * *
----->-----
*      CUST_ID           INTEGER        4          0 N      N      1 INSERT
*      CATEGORY          CHAR            2          0 N      N      2 INSERT
*      DEDUCTABLE        DECIMAL        15          2 N      N      3 INSERT
*      COVERAGE          DECIMAL        15          2 N      N      4 INSERT
*      SYS_STA           TIMESTMP       13         12 N      Q      5 UPDATE
*      SYS_END           TIMESTMP       13         12 N      R      6 UPDATE
*      BUS_STA           DATE            4          0 N      N      7 INSERT
*      BUS_END           DATE            4          0 N      N      8 INSERT
*      CREATE_ID         TIMESTMP       13         12 N      X      9 UPDATE
***** END OF DB2 DATA *****

```

Figure 12-14 Operation type of CREATE_ID is changed to UPDATE

All the column definitions for the CUST_COVERAGE table are now completed. Now you need to define the SYSTEM_TIME period and BUSINESS_TIME period.

12.3.1 TBLOPTS command

There is a new command in DB2 Administration Tool V10 called the TBLOPTS command. This command can be found on the Create Table Columns panel (ADB26CTF). The TBLOPTS command opens the Create Table Options panel (ADB26TOP), which allows you to specify options such as the EDITPROC, VALIDPROC, AUDIT, CAPTURE, and VOLATILE parameters for the table.

The ADB26TOP panel is where you specify whether you want to define the SYSTEM_TIME period or BUSINESS_TIME period for the table you are creating. Enter TBLOPTS on the Command line or place the cursor under the TBLOPTS command and press Enter (Figure 12-15).

```

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==> TBLOPTS                               Scroll ==> CSR

Schema . . ADMR2      >           Database . . .
Name . . . CUST_COVERAGE      > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands:  M - Move  A - After  B - Before
                Inn - Insert  U - Update  D - Delete  Rnn - Repeat

                                Operation
Select Column Name           Col Type           Length Scale  Null D Col No Type
      *                   *                   *      * *      *      * *
----->-----

*      CUST_ID                INTEGER                4          0 N      N      1 INSERT
*      CATEGORY               CHAR                   2          0 N      N      2 INSERT
*      DEDUCTABLE              DECIMAL                15         2 N      N      3 INSERT
*      COVERAGE                DECIMAL                15         2 N      N      4 INSERT
*      SYS_STA                 TIMESTMP               13        12 N      Q      5 UPDATE
*      SYS_END                 TIMESTMP               13        12 N      R      6 UPDATE
*      BUS_STA                 DATE                   4          0 N      N      7 INSERT
*      BUS_END                 DATE                   4          0 N      N      8 INSERT
*      CREATE_ID               TIMESTMP               13        12 N      X      9 UPDATE

***** END OF DB2 DATA *****

```

Figure 12-15 TBLOPTS command


```
ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9  
Command ==> TBLOPTS Scroll ==> CSR
```



```
Sc E sssssssssssssssssssssssssssssssssssssssssssssssssssssssN  
Na e ADB26TOP ----- VA1A Create Table Options ----- e  
e e e  
Co e Enter values and press enter. e  
Li e e  
I e EDITPROC . . . . . e  
e VALIDPROC . . . . . e n  
Se e AUDIT . . . . . (None, Changes, or All) e  
e DATA CAPTURE . . . NONE (None/Changes) e  
-- e CCSID . . . . . (EBCDIC, UNICODE, ASCII) e -  
e RESTRICT ON DROP . . (Yes/No) e  
* e VOLATILE . . . . . (Yes/No) e  
* e APPEND . . . . . (Yes/No) e  
* e PBG size . . . . . (in GB) e  
* e PBR . . . . . (Yes/No) e  
* e System period . . . YES (Yes/No) e  
* e Business period . . . YES (Yes/No) e  
* e e e  
* D sssssssssssssssssssssssssssssssssssssssssssssssssssssM  
* CREATE_ID TIMESTMP 13 12 N X 9 UPDATE  
***** END OF DB2 DATA *****
```

When YES is specified for the System period, the ROW BEGIN and ROW END columns of the table are used as the Start and End columns of the SYSTEM_TIME period. In this case, it would be the SYS_STA and SYS_END columns.

On the ADBP26CTB panel, you use the S and E line commands to specify the columns to be used as the Starting time column and Ending time column for the BUSINESS_TIME period. In the case where more than two columns can be selected as the starting or ending time columns for the BUSINESS_TIME period, the R- remove line command is used to remove the Start or End setting for the column. Let us look at a case in which multiple columns are eligible to define the BUSINESS_TIME period before continuing our Create Table example.

Another case: Multiple columns eligible for BUSINESS_TIME period

In the following case, there are four columns, B_1, O_1, B_2, and O_2, which can be defined as the starting and ending time columns for BUSINESS_TIME period. First, select the B_1 as the starting time column and B_2 as the ending time column by issuing the S line command, which is entered next to the B_1, and the E line command, which entered is next to the B_2 column (Figure 12-17).

```
ADBP6CTB ----- VA1A Select BUSINESS TIME Period Columns Row 1 to 4 of 4
Command ==>                                     Scroll ==> CSR

Schema . . . >
Name . . . TEMP01 >

Commands: CANCEL
Line commands: S - Start column E - End column R - Remove

Select Column Name      Col Type Length Scale Null Def Col No Column
      *                *      *      *      *   *   *
----->----->----->----->----->----->----->----->----->
s      B_1              DATE              0 N    N      2
      O_1              DATE              0 N    N      3
e      B_2              DATE              0 N    N      4
      O_2              DATE              0 N    N      5
***** END OF DB2 DATA *****
```

Figure 12-17 Specify the starting and ending column from multiple columns

After pressing Enter, the Period Column of the B_1 and B_2 column is updated to reflect the Starting and Ending columns (Figure 12-18).

```
ADBP6CTB ----- VA1A Select BUSINESS TIME Period Columns Row 1 to 4 of 4
Command ==>                                     Scroll ==> CSR

Schema . . . >
Name . . . TEMP01 >

Commands: CANCEL
Line commands: S - Start column E - End column R - Remove

Select Column Name      Col Type Length Scale Null Def Col No Column
      *                *      *      *      *   *   *
----->----->----->----->----->----->----->----->
*      B_1              DATE              0 N    N      2 Start
      O_1              DATE              0 N    N      3
*      B_2              DATE              0 N    N      4 End
      O_2              DATE              0 N    N      5
***** END OF DB2 DATA *****
```

Figure 12-18 Period Column updated to reflect the B_1 and B_2 columns

Now you want to change the Ending time column from B_2 to O_1. First, perform a REMOVE on the B_2 in the ending column and issue the E line command next to the O_1 column (Figure 12-19).

```
ADBP6CTB ----- VA1A Select BUSINESS TIME Period Columns Row 1 to 4 of 4
Command ==> Scroll ==> CSR

Schema . . . >
Name . . . TEMP01 >

Commands: CANCEL
Line commands: S - Start column E - End column R - Remove
```

Select	Column Name	Col Type	Length	Scale	Null	Def	Col No	Period Column
*	*	*	*	*	*	*	*	
	B_1	DATE			0 N	N	2	Start
e	O_1	DATE			0 N	N	3	
r	B_2	DATE			0 N	N	4	End
	O_2	DATE			0 N	N	5	

***** END OF DB2 DATA *****

Figure 12-19 Respecifying the ending time column

After pressing Enter, the Only one start and one end column are allowed message appears under the Command line (Figure 12-20). This message appears because the line command on the ADBP6CTB panel is been processed sequentially and the E(nd column) line command was encountered before the R(emove) line command. Thus, the E line command failed due to the fact that the ending time column has already been defined (as B_2). If the E line command was entered after the R line command, for example, the E line command was issued against O_2 column, both the E and R line commands are processed successfully after pressing Enter.

```
ADBP6CTB ----- VA1A Select BUSINESS TIME Period Columns Row 1 to 4 of 4
Command ==> Scroll ==> CSR
Only one start and one end column are allowed.
Schema . . . >
Name . . . TEMP01 >

Commands: CANCEL
Line commands: S - Start column E - End column R - Remove
```

Select	Column Name	Col Type	Length	Scale	Null	Def	Col No	Period Column
*	*	*	*	*	*	*	*	
*	B_1	DATE			0 N	N	2	Start
E	O_1	DATE			0 N	N	3	
R	B_2	DATE			0 N	N	4	End
	O_2	DATE			0 N	N	5	

***** END OF DB2 DATA *****

Figure 12-20 Error message displayed on the ADBP6CTB panel

In this case, you need to first issue the R line command against the B_2 column and press Enter. After the R command is executed successfully, the End state from the period column of the B_2 column is cleared (Figure 12-21).

```

ADBP6CTB ----- VA1A Select BUSINESS TIME Period Columns Row 1 to 4 of 4
Command ==>                                         Scroll ==> CSR

Schema . . . >
Name . . . TEMP01 >

Commands: CANCEL
Line commands: S - Start column E - End column R - Remove

Select Column Name      Col Type  Length Scale  Null Def Col No  Period
      *                *      *      *      *  *  *
----->-----
*      B_1              DATE              0 N    N      2 Start
      O_1              DATE              0 N    N      3
*      B_2              DATE              0 N    N      4
      O_2              DATE              0 N    N      5
***** END OF DB2 DATA *****

```

Figure 12-21 End Period Column state was cleared from B_2 column

Now we can enter the E line command next to the O_1 column and select the O_1 column as the ending time column (Figure 12-22).

```

ADBP6CTB ----- VA1A Select BUSINESS TIME Period Columns Row 1 to 4 of 4
Command ==>                                         Scroll ==> CSR

Schema . . . >
Name . . . TEMP01 >

Commands: CANCEL
Line commands: S - Start column E - End column R - Remove

Select Column Name      Col Type  Length Scale  Null Def Col No  Period
      *                *      *      *      *  *  *
----->-----
*      B_1              DATE              0 N    N      2 Start
e      O_1              DATE              0 N    N      3
*      B_2              DATE              0 N    N      4
      O_2              DATE              0 N    N      5
***** END OF DB2 DATA *****

```

Figure 12-22 Enter the E line command next to the O_1 column

After pressing Enter, the Period Column of O_1 is changed to End, which indicates that the O_1 column is the Ending time column for the BUSINESS_TIME period (Figure 12-26 on page 328).

```

ADBP6CTB ----- VA1A Select BUSINESS TIME Period Columns Row 1 to 4 of 4
Command ==>                                         Scroll ==> CSR

Schema . . . >
Name . . . TEMP01 >

Commands: CANCEL
Line commands: S - Start column E - End column R - Remove

Select Column Name      Col Type  Length Scale  Null Def Col No  Period
      *                *      *      *      *   *   *
----->-----
*      B_1              DATE              0 N    N      2  Start
*      O_1              DATE              0 N    N      3  End
*      B_2              DATE              0 N    N      4
      O_2              DATE              0 N    N      5
***** END OF DB2 DATA *****

```

Figure 12-23 Period Column of O_1 changed to End

Continuing the creation of a bi-temporal table using the CT function

Let us continue creating CUST_COVERAGE, which is a bi-temporal table, using the DB2 Administration Tool Create Table (CT) function.

In this case, there are only two columns that can be used to define the BUSINESS_TIME period. Both the BUS_STA and BUS_END columns are displayed on the BUSINESS TIME Period Column panel (ADBP6CTB) (Figure 12-24).

```

ADBP6CTB ----- VA1A Select BUSINESS TIME Period Columns Row 1 to 2 of 2
Command ==>                                         Scroll ==> CSR

Schema . . . ADMR2 >
Name . . . CUST_COVERAGE >

Commands: CANCEL
Line commands: S - Start column E - End column R - Remove

Select Column Name      Col Type  Length Scale  Null Def Col No  Period
      *                *      *      *      *   *   *
----->-----
      BUS_STA           DATE              0 N    N      7
      BUS_END           DATE              0 N    N      8
***** END OF DB2 DATA *****

```

Figure 12-24 Select BUSINESS TIME Period Columns Row

Use the S and E line commands to specify the columns to be used as the Starting time column and Ending time column for the BUSINESS_TIME period. Enter S next to the BUS_STA column and E next to the BUS_END column (Figure 12-25) and press Enter.

```
ADBP6CTB ----- VA1A Select BUSINESS TIME Period Columns Row 1 to 2 of 2
Command ==>                                     Scroll ==> CSR

Schema . . ADMR2      >
Name . . . CUST_COVERAGE      >

Commands: CANCEL
Line commands: S - Start column E - End column R - Remove

Select Column Name      Col Type  Length Scale  Null Def Col No Column
      *                *      *      *      *   *   *
----->-----
S      BUS_STA          DATE              0 N    N      7
E      BUS_END          DATE              0 N    N      8
***** END OF DB2 DATA *****
```

Figure 12-25 Specifying the Starting and Ending time column

After pressing Enter, the period column field is updated to reflect the starting time column and the ending time column. Notice that the Period Column of the BUS_STA is updated to Start and the period column for the BUS_END column is changed to End (Figure 12-26).

```
ADBP6CTB ----- VA1A Select BUSINESS TIME Period Columns Row 1 to 2 of 2
Command ==>                                     Scroll ==> CSR

Schema . . ADMR2      >
Name . . . CUST_COVERAGE      >

Commands: CANCEL
Line commands: S - Start column E - End column R - Remove

Select Column Name      Col Type  Length Scale  Null Def Col No Column
      *                *      *      *      *   *   *
----->-----
*      BUS_STA          DATE              0 N    N      7 Start
*      BUS_END          DATE              0 N    N      8 End
***** END OF DB2 DATA *****
```

Figure 12-26 Period Column updated for the starting and ending time column

Enter the END command on Select BUSINESS TIME Period panel (ADBP6CTB) or press PF3 after you finish defining the Starting and Ending time column for the BUSINESS_TIME period. You return to the Create Table Columns panel (ADB26CTF).

Enter CREATE at the Command line or place the cursor under the CREATE command line on Create Table Columns panel (ADB26CTF) to issue the Create table statement (Figure 12-27).

```

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==> CREATE                               Scroll ==> CSR

Schema . . ADMR2      >          Database . . .
Name . . . CUST_COVERAGE      > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands: M - Move  A - After  B - Before
               Inn - Insert  U - Update  D - Delete  Rnn - Repeat

Select Column Name      Col Type      Length Scale  Null D Col No Type
      *                  *            *      * *      *
-----> -----
*      CUST_ID           INTEGER        4          0 N      N      1 INSERT
*      CATEGORY          CHAR            2          0 N      N      2 INSERT
*      DEDUCTABLE        DECIMAL        15          2 N      N      3 INSERT
*      COVERAGE          DECIMAL        15          2 N      N      4 INSERT
*      SYS_STA            TIMESTMP       13         12 N      Q      5 UPDATE
*      SYS_END            TIMESTMP       13         12 N      R      6 UPDATE
*      BUS_STA            DATE            4          0 N      N      7 INSERT
*      BUS_END            DATE            4          0 N      N      8 INSERT
*      CREATE_ID          TIMESTMP       13         12 N      X      9 UPDATE
***** END OF DB2 DATA *****

```

Figure 12-27 Issue the Create command to create a table

After pressing Enter, the CREATE stmt executed message is displayed under the Command line, which indicates that the Create Table statement for CUST_COVERAGE table has been successfully executed (Figure 12-28).

```
ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 9 of 9
Command ==>                                         Scroll ==> CSR
CREATE stmt executed
Schema . . ADMR2      >      Database . . .
Name . . . CUST_COVERAGE      > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands: M - Move A - After B - Before
Inn - Insert U - Update D - Delete Rnn - Repeat

Select Column Name      Col Type      Length Scale Null D Col No Type
      *                  *            *      * *      *      * *
----->-----
*      CUST_ID           INTEGER        4      0 N      N      1 INSERT
*      CATEGORY          CHAR            2      0 N      N      2 INSERT
*      DEDUCTABLE        DECIMAL         15      2 N      N      3 INSERT
*      COVERAGE          DECIMAL         15      2 N      N      4 INSERT
*      SYS_STA           TIMESTMP        13     12 N      Q      5 UPDATE
*      SYS_END           TIMESTMP        13     12 N      R      6 UPDATE
*      BUS_STA           DATE             4      0 N      N      7 INSERT
*      BUS_END           DATE             4      0 N      N      8 INSERT
*      CREATE_ID         TIMESTMP        13     12 N      X      9 UPDATE
***** END OF DB2 DATA *****
```

Figure 12-28 TABLE CUST_COVERAGE created

Press PF3 four times to return to the DB2 Administration Menu panel (ADB2). To verify that the CUST_COVERAGE bi-temporal table has been created successfully, go to the system catalog and search for the ADMR2.CUST_COVERAGE table. Specify 1 at the Option line on the DB2 Administration Menu panel (ADB2) and press Enter (Figure 12-29).

```
ADB2 dmin ----- DB2 Administration Menu 10.1.0 -----
Option ==> 1

      1 - DB2 system catalog                DB2 System: VA1A
      2 - Execute SQL statements            DB2 SQL ID: ADMR2
      3 - DB2 performance queries          Userid   : SYSADM
      4 - Change current SQL ID             DB2 Schema: ADMR2
      5 - Utility generation using LISTDEFS and TEMPLATES DB2 Rel   : 1015
      P - Change DB2 Admin parameters
      DD - Distributed DB2 systems
      E - Explain
      Z - DB2 system administration
      SM - Space management functions
      W - Manage work statement lists
      X - Exit DB2 Admin
      CC - DB2 catalog copy version maintenance
      CM - Change management

                                                    More:      +

Interface to other DB2 products and offerings:
      I  DB2I

      C  DB2 Object Comparison Tool
```

Figure 12-29 Back to the DB2 Administration Menu

After pressing Enter, the System Catalog panel (ADB21) opens. Specify CUST_COVERAGE as the name and AMDR2 as the owner. Then enter TT at the Option line (Figure 12-30). TT is a short cut for a look up table with the type of T. Thus, it eliminates displaying the table that is a View, Alias, MQT or History table (see 12.4.4, “History table” on page 341).

```

ADB21 min ----- VA1A System Catalog -----
Option ==> tt

                                     More:      +
Object options:                      DB2 System: VA1A
      AO - Authorization options      DB2 SQL ID: AMDR2
      G - Storage groups              P - Plans
      D - Databases                   L - Collections
      S - Table spaces                K - Packages
      T - Tables, views, and aliases M - DBRMs
      V - Views                       H - Schemas
      A - Aliases                     E - User defined data types
      Y - Synonyms                    F - Functions
      X - Indexes                     O - Stored procedures
      C - Columns                     J - Triggers
      N - Constraints                 Q - Sequences
      DS - Database structures         DSP - DS with plans and packages
      PDC - DB2 pending definition changes

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name      ==> cust_coverage      > Grantor ==>      >
Owner     ==> AMDR2      >      Grantee ==>      >
In D/L/H ==>      > Switch Catalog Copy ==> N      (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)

```

Figure 12-30 Issue TT command to locate the AMDR2.CUST_COVERAGE table

After pressing Enter, the CUST_COVERAGE table with Schema name AMDR2 opens on the Tables, Views, and Aliases panel (ADB21T). To examine the contents of the AMDR2.CUST_COVERAGE table, issue the DDL line command next to the CUST_COVERAGE table and check the generated DDL (Figure 12-31).

```

ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==>                                         Scroll ==> CSR

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel   Name              Schema  T DB Name  TS Name   Cols      Rows ChksC
      *                *      *  *      *      *        *    **
-----
ddl   CUST_COVERAGE     AMDR2   T DSN00015 CUSTRCOV    9         -1     1
***** END OF DB2 DATA *****

```

Figure 12-31 Issue DDL line command against CUST_COVERAGE table

After pressing Enter, the generated DDL of the CUST_COVERAGE table opens in an ISPF edit session (Figure 12-32). The CUST_COVERAGE table is a bi-temporal table because there are both BUSINESS_TIME period and SYSTEM_TIME periods defined on the table.

```

ISREDDE2  SYS10325.T174925.RA000.SYSADM.R0100063          Columns 00001 00072
Command ==>                                         Scroll ==> CSR
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>         your edit profile using the command RECOVERY ON.
000001  SET CURRENT SQLID='ADMR2';
000002  CREATE TABLE ADMR2.CUST_COVERAGE
000003      (CUST_ID          INTEGER NOT NULL,
000004      CATEGORY          CHAR(2) FOR SBCS DATA NOT NULL,
000005      DEDUCTABLE          DECIMAL(15, 2) NOT NULL,
000006      COVERAGE          DECIMAL(15, 2) NOT NULL,
000007      SYS_STA           TIMESTAMP (12) WITHOUT TIME ZONE NOT NULL
000008          GENERATED ALWAYS AS ROW BEGIN,
000009      SYS_END            TIMESTAMP (12) WITHOUT TIME ZONE NOT NULL
000010          GENERATED ALWAYS AS ROW END,
000011      BUS_STA           DATE NOT NULL,
000012      BUS_END           DATE NOT NULL,
000013      CREATE_ID          TIMESTAMP (12) WITHOUT TIME ZONE NOT NULL
000014          GENERATED ALWAYS AS TRANSACTION START ID,
000015      PERIOD BUSINESS_TIME (BUS_STA, BUS_END),
000016      PERIOD SYSTEM_TIME (SYS_STA, SYS_END))
000017  PARTITION BY SIZE EVERY 4 G
000018  AUDIT NONE
000019  DATA CAPTURE NONE
000020  CCSID      EBCDIC
000021  NOT VOLATILE
000022  APPEND NO ;
000023  COMMIT;
***** ***** Bottom of Data *****

```

Figure 12-32 Generated DDL for CUST_COVERAGE

After the temporal table is created, to take advantage of the data versioning, you still need to create the associated history table (refer to 12.4.4, “History table” on page 341) and add versioning (refer to 12.4.6, “ADD VERSIONING option” on page 346) by creating the link between the base temporal table and its history table.

Creating the associated history table and how to add versioning to the temporal table are described in 12.4, “Using the AL line command to convert a table to a temporal table” on page 333.

12.4 Using the AL line command to convert a table to a temporal table

In the following example, we show you the step by step instructions to change an existing table, CUSTOMER_COVERAGE, to a system temporal table. A history table is created and added to the table to add versioning. All prior rows from the changed rows that resulted from the SQL UPDATE and DELETE process are moved to the associated history table.

12.4.1 CUSTOMER_COVERAGE table

The CUSTOMER_COVERAGE table contains give rows of data. The DDL of the CUSTOMER_COVERAGE table is shown in Figure 12-33.

```
SET CURRENT SQLID='ADMR2';
CREATE TABLE ADMR2.CUSTOMER_COVERAGE
  (CUST_ID          INTEGER NOT NULL,
   CATEGORY        CHAR(2) FOR SBCS DATA NOT NULL,
   DEDUCTABLE      DECIMAL(15, 2) NOT NULL,
   COVERAGE        DECIMAL(15, 2) NOT NULL,
   CONSTRAINT CUST_ID
   PRIMARY KEY (CUST_ID,
               CATEGORY))
PARTITION BY SIZE EVERY 4 G
AUDIT NONE
DATA CAPTURE NONE
CCSID      EBCDIC
NOT VOLATILE
APPEND NO  ;
COMMIT;
```

Figure 12-33 DDL of CUSTOMER_COVERAGE table

The data with the table is shown in Figure 12-34.

```
ADB2DF in -- VA1A LIST ADMR2.CUSTOMER_COVERAGE          ----- Row 1 to 5 of 5
Command ==>                                           Scroll ==> CSR

L      CUST_ID CATEGORY          DEDUCTABLE          COVERAGE
      * *                      *                      *
- - - - -
      111111 01                500.00          10000.00
      222222 01                500.00          20000.00
      333333 02               1000.00          30000.00
      444444 01               1000.00          40000.00
      555555 02               1000.00          35000.00
***** END OF DB2 DATA *****
```

Figure 12-34 Five rows of data in CUSTOMER_COVERAGE table

To create a SYSTEM_TIME temporal table, you need to define the following three columns:

- Beginning SYSTEM time column

This is a column that is maintained by DB2. The column is defined as GENERATED ALWAYS AS ROW BEGIN. The data type of the column needs to be TIMESTAMP(12) NOT NULL WITHOUT TIMEZONE.

- Ending SYSTEM time column

This a column that is maintained by DB2. The column is defined as GENERATED ALWAYS AS ROW END. The data type of the column needs to be TIMESTAMP(12) NOT NULL WITHOUT TIMEZONE

► Transaction Start ID column

This a column that is maintained by DB2. The column is defined as GENERATED ALWAYS AS TRANSACTION START ID. The data type of the column needs to be TIMESTAMP(12) WITHOUT TIMEZONE.

In addition to the above three columns, you also need to define the SYSTEM_Time PERIOD, which specifies the beginning and ending system time column.

We show how to change the CUSTOMER_COVERAGE table to a SYSTEM TIME temporal table using the Administration Tool V10 AL command in the following sections.

First, issue the AL line command next to the CUSTOMER_COVERAGE table (Figure 12-35).

```
ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel  Name          Schema  T DB Name  TS Name  Cols  Rows Chks C
*      *            *    *      *      *      *    * *  *
-----
AL    CUSTOMER_COVERAGE  ADMR2   T DSN00005  CUSTOMER    4      -1    0
***** END OF DB2 DATA *****
```

Figure 12-35 Issue AL line command

Press Enter. The Alter Table panel (ADB21TA) opens (Figure 12-36 on page 336). There are several new options that have been added to ADB21TA for DB2 10 for z/OS support. We focus on the ADD column, ADD PERIOD, ADD VERSIONING, and DROP VERSIONING options, which are related to the temporal table support.

12.4.2 ADD column option

The ADD column option has been enhanced to allow you add the column that is required to create the SYTEM_TIME period and complete the SYSTEM_TIME temporal table definition. There are three columns that are needed to define the SYSTEM-TIME period temporal table:

- The beginning SYSTEM time column
- The ending SYSTEM time column
- The transaction start ID column

We use the ADD column option to specify the three columns, one at a time.

Enter S next to the ADD column option on the Alter Table panel (ADB21TA) (Figure 12-36).

```

ADB21TA n ----- VA1A Alter Table ----- 18:24
Command ==>

Table schema . . . : ADMR2      >
Table name . . . : CUSTOMER_COVERAGE  >

AUDIT . . . . . NONE          (None, Changes, or All)
DATA CAPTURE . . . . . NONE    (None/Changes)
VALIDPROC . . . . . NULL      (NULL/Program name)
RESTRICT ON DROP . . . NO      (Yes/No)
VOLATILE . . . . . NO          (Yes/No)
APPEND . . . . . NO            (Yes/No)

ALTER TABLE with any of the above changes OR select one of the options below

s ADD column                                ADD MATERIALIZED QUERY
ADD PRIMARY KEY                            DROP MATERIALIZED QUERY
DROP PRIMARY KEY                          REFRESH MATERIALIZED TABLE
ADD FOREIGN KEY                           ADD PARTITIONING KEY
DROP FOREIGN KEY                          ADD PARTITION
ADD CHECK constraint                       ADD CLONE
DROP CHECK constraint                     DROP CLONE
ADD UNIQUE constraint                     ADD VERSIONING
DROP UNIQUE constraint                   DROP VERSIONING
ADD PERIOD                                ACTIVATE COLUMN ACCESS CONTROL
ACTIVATE ROW ACCESS CONTROL              DEACTIVATE COLUMN ACCESS CONTROL
DEACTIVATE ROW ACCESS CONTROL

```

Figure 12-36 Select the ADD column option

Press Enter. The Alter Table panel (ADB21TAB) opens. You can specify the definition for the begin time column. The begin time column contains the timestamp value for when a row is created. The column is defined with the data type of `TIMESTAMP(12) WITHOUT TIME ZONE AND NOT NULL` with the `GENERATED Q - ALWAYS AS ROW BEGIN`, and the data type cannot be a user-defined type (Figure 12-37).

ADB21TAB ----- VA1A Alter Table ----- 1

Command ==>

More: +

ALTER TABLE

Table schema . . ADMR2 >

Table name . . . CUSTOMER_COVERAGE >

ADD

Column name . . SYS_STA > (? to look up)

Column type . . TIMESTAMP (Built-in only)

Data length . . (Built-in only)

Inline length . . (0-32680 BLOB or CLOB, 0-16340 DBCLOB)

Precision . . . (used only w/FLOAT and DECIMAL)

Scale 12 (used only w/DECIMAL and TIMESTAMP)

Type schema . . > (User-defined only)

Type name . . . > (User-defined only)

WITH TIME ZONE . NO (Yes/No - for TIMESTAMP only)

Allow nulls . . NO (Yes or blank-nullable, No-NOT NULL)

FOR ? DATA . . . (B-Bit, S-SBCS, M-Mixed, blank-N/A)

WITH DEFAULT . . (Yes, No, L (SECLABEL) or enter value below)

Default value . . . >

GENERATED . . . Q (A-ALWAYS, D-DEFAULT, I-ALWAYS AS IDENTITY, J-DEFAULT AS IDENTITY, E-ALWAYS AS UPD TIMESTAMP, F-DEFAULT AS UPD TIMESTAMP, Q-ALWAYS AS ROW BEGIN, R-ALWAYS AS ROW END, X-ALWAYS AS TRANSACTION START ID)

FIELDPROC

Program name . . (optional)

Program parm . . . >

Hidden (Yes/No)

Figure 12-37 ADD system begin time column

Press Enter. The ALTER stmt executed message opens, which means the begin time column has been successfully added to the table, provided that the prompt option is not on and there is no change management specified for this ALTER.

Next, you need to specify the definition for the time ending column. The end time column contains the timestamp value for when a row is updated or deleted. The column is defined with the data type of `TIMESTAMP(12) WITHOUT TIME ZONE AND NOT NULL` with the `GENERATED R - ALWAYS AS ROW END`, and the data type cannot be a user-defined type. (Figure 12-38).

ADB21TAB ----- VA1A Alter Table -----

Command ==>

More: +

ALTER TABLE

Table schema . . ADMR2 >

Table name . . . CUSTOMER_COVERAGE >

ADD

Column name . . SYS_END > (? to look up)

Column type TIMESTAMP (Built-in only)

Data length (Built-in only)

Inline length (0-32680 BLOB or CLOB, 0-16340 DBCLOB)

Precision (used only w/FLOAT and DECIMAL)

Scale 12 (used only w/DECIMAL and TIMESTAMP)

Type schema > (User-defined only)

Type name > (User-defined only)

WITH TIME ZONE . NO (Yes/No - for TIMESTAMP only)

Allow nulls . . . NO (Yes or blank-nullable, No-NOT NULL)

FOR ? DATA . . . (B-Bit, S-SBCS, M-Mixed, blank-N/A)

WITH DEFAULT . . . (Yes, No, L (SECLABEL) or enter value below)

Default value >

GENERATED . . . R (A-ALWAYS, D-DEFAULT, I-ALWAYS AS IDENTITY, J-DEFAULT AS IDENTITY, E-ALWAYS AS UPD TIMESTAMP, F-DEFAULT AS UPD TIMESTAMP, Q-ALWAYS AS ROW BEGIN, R-ALWAYS AS ROW END, X-ALWAYS AS TRANSACTION START ID)

FIELDPROC

Program name (optional)

Program parm >

Hidden (Yes/No)

Figure 12-38 ADD system end time column

Press Enter. The ALTER stmt executed message opens, which means the end time column is successfully added to the table, provided that the prompt option is not on and there is no change management specified for this ALTER.

Next, you need to specify the definition for the transaction start ID column. The transaction ID column is defined with the data type of `TIMESTAMP(12) WITHOUT TIME ZONE` and `GENERATED X - ALWAYS AS TRANSACTION START ID`, and the data type cannot be a user-defined type (Figure 12-39).

ADB21TAB ----- VA1A Alter Table -----

Command ==>

More: +

ALTER TABLE

Table schema . . ADMR2 >

Table name . . . CUSTOMER_COVERAGE >

ADD

Column name . . CREATE_ID > (? to look up)

Column type . . TIMESTAMP (Built-in only)

Data length . . (Built-in only)

Inline length . (0-32680 BLOB or CLOB, 0-16340 DBCLOB)

Precision . . . (used only w/FLOAT and DECIMAL)

Scale 12 (used only w/DECIMAL and TIMESTAMP)

Type schema . . > (User-defined only)

Type name . . . > (User-defined only)

WITH TIME ZONE . NO (Yes/No - for TIMESTAMP only)

Allow nulls . . (Yes or blank-nullable, No-NOT NULL)

FOR ? DATA . . . (B-Bit, S-SBCS, M-Mixed, blank-N/A)

WITH DEFAULT . . (Yes, No, L (SECLABEL) or enter value below)

Default value . >

GENERATED . . . X (A-ALWAYS, D-DEFAULT, I-ALWAYS AS IDENTITY, J-DEFAULT AS IDENTITY, E-ALWAYS AS UPD TIMESTAMP, F-DEFAULT AS UPD TIMESTAMP, Q-ALWAYS AS ROW BEGIN, R-ALWAYS AS ROW END, X-ALWAYS AS TRANSACTION START ID)

FIELDPROC

Program name . . (optional)

Program parm . . >

Hidden (Yes/No)

Figure 12-39 ADD transaction start ID column

Press Enter. The ALTER stmt executed message opens, which indicates that the transaction start ID column is successfully added to the table, provided that the prompt option is not on and there is no change management specified for this ALTER.

Now that you have successfully added the three columns that are required for the `SYSTEM_TIME` temporal table, you need to create the `SYSTEM_TIME` period.

Press PF3, and you return to Alter Table panel (ADB21TA).

12.4.3 ADD PERIOD option

ADD PERIOD is a DB2 Administration Tool V10 feature that allows you to ADD a business time or system time period to a table. You need to specify the time begin column and the time end column for the period you are creating. In our CUSTOMER_COVERAGE table example, we add the SYSTEM_TIME period only.

Specify s next to the ADD PERIOD option on Alter Table panel (ADB21TA) (Figure 12-40).

```
ADB21TA n ----- VA1A Alter Table -----
Command ==>

Table schema . . : ADMR2      >
Table name . . . : CUSTOMER_COVERAGE >

AUDIT . . . . . NONE      (None, Changes, or All)
DATA CAPTURE . . . . . NONE (None/Changes)
VALIDPROC . . . . . NULL   (NULL/Program name)
RESTRICT ON DROP . . . NO   (Yes/No)
VOLATILE . . . . . NO      (Yes/No)
APPEND . . . . . NO        (Yes/No)

ALTER TABLE with any of the above changes OR select one of the options below

DROP FOREIGN KEY          ADD PARTITION
ADD CHECK constraint      ADD CLONE
DROP CHECK constraint     DROP CLONE
ADD UNIQUE constraint      ADD VERSIONING
DROP UNIQUE constraint    DROP VERSIONING
s ADD PERIOD              ACTIVATE COLUMN ACCESS CONTROL
ACTIVATE ROW ACCESS CONTROL DEACTIVATE COLUMN ACCESS CONTROL
DEACTIVATE ROW ACCESS CONTROL
```

Figure 12-40 ADD PERIOD option

Press Enter. The Add Period panel (ADB21TAP) opens.

Specify the type of period (S for SYSTEM TIME or B for BUSINESS TIME) and the start time column and the end time columns. The columns must exist in the table with the correct begin time and end time definition (Figure 12-41).

```
ADBPTAP n ----- VA1A Add Period -----
Command ==>

ALTER TABLE "ADMR2"."CUSTOMER_COVERAGE"
ADD PERIOD

Type . . . . . S      (S-SYSTEM TIME or B-BUSINESS TIME)

Start column name . . SYS_STA      (? to lookup)
End column name . . . SYS_END      (? to lookup)
```

Figure 12-41 ADD PERIOD SYSTEM_Time

Press Enter. The ALTER stmt. executed message opens, which indicates that the SYSTEM_TIME period has been successfully added to the table.

Press F3, and you will be returned to the Alter Table panel (ADB21TA).

12.4.4 History table

The SYSTEM_TIME period is meaningful because of versioning. Versioning means that the old rows are archived into a separate table. The table that contains the current active rows of data is called the system period temporal table. The table that contains the archived rows is called the history table. When you define versioning on a system period temporal table, you must create a corresponding history table.

A history table must have the same number of columns, the same names, data types, null attributes, CCSIDs, subtypes, hidden attributes, and field procedures as the corresponding system period temporal table.

If the system period temporal table has a column of ROWID GENERATED ALWAYS or ROWID GENERATED BY DEFAULT, the history table must have a corresponding ROWID GENERATED ALWAYS column. Otherwise, the history table should not have any GENERATED columns.

There are different ways to create a table. You can specify option 2.4 on the DB2 Administration Tool main menu and use the CT (Create Table) function to create the history table to match the system period temporal table. In our example, we use the DDL line command to create a history table that corresponds to the newly modified CUSTOMER_COVERAGE table,

12.4.5 Using the DDL line command to create a history table

On the System Catalog panel (ADB21), specify T (Tables) as the option and CUSTOMER_COVERAGE as the object name (Figure 12-42).

```
ADB21 min ----- VA1A System Catalog -----
Option ==> T

Object options:
  AO - Authorization options
  G - Storage groups
  D - Databases
  S - Table spaces
  T - Tables, views, and aliases
  V - Views
  A - Aliases
  Y - Synonyms
  X - Indexes
  C - Columns
  N - Constraints
  DS - Database structures
  PDC - DB2 pending definition changes
  P - Plans
  L - Collections
  K - Packages
  M - DBRMs
  H - Schemas
  E - User defined data types
  F - Functions
  O - Stored procedures
  J - Triggers
  Q - Sequences
  DSP - DS with plans and packages

More: +
DB2 System: VA1A
DB2 SQL ID: SYSADM

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name ==> CUSTOMER_COVERAGE > Grantor ==> >
Owner ==> ADMR2 > Grantee ==> >
In D/L/H ==> > Switch Catalog Copy ==> N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
```

Figure 12-42 Locate CUSTOMER_COVERAGE table

Press Enter. The Tables, Views, and Aliases panel (ADB21T) opens. Enter the DDL line command next to the CUSTOMER_COVERAGE table (Figure 12-43).

```
ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel Name          Schema  T DB Name  TS Name  Cols  Rows  Chks  C
*                *      * *         *        *      *    *  *
-----
ddl  CUSTOMER_COVERAGE  ADMR2    T DSN00005  CUSTOMER      7      -1    0
***** END OF DB2 DATA *****
```

Figure 12-43 Issue the DDL line command to the CUSTOMER_COVERAGE table

Press Enter. The DDL of the customer_coverage table is extracted and placed in an edit session (Figure 12-44). Because the history table of the CUSTOMER_COVERAGE table needs to have the same number of columns, names, and data types without the GENERATED ALWAYS attribute, we modify the DDL of the CUSTOMER_COVERAGE table and use it to create its corresponding history table.

```

ISREDDE2   SYS10307.T154722.RA000.SYSADM.R0100184           Columns 00001 00072
Command ==>                                           Scroll ==> CSR
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001    SET CURRENT SQLID='ADMR2';
000002    CREATE TABLE ADMR2.CUSTOMER_COVERAGE
000003        (CUST_ID              INTEGER NOT NULL,
000004         CATEGORY                CHAR(2) FOR SBCS DATA NOT NULL,
000005         DEDUCTABLE              DECIMAL(15, 2) NOT NULL,
000006         COVERAGE               DECIMAL(15, 2) NOT NULL,
000007         SYS_STA                TIMESTAMP (12) WITHOUT TIME ZONE NOT NULL
000008         GENERATED ALWAYS AS ROW BEGIN,
000009         SYS_END                TIMESTAMP (12) WITHOUT TIME ZONE NOT NULL
000010         GENERATED ALWAYS AS ROW END,
000011         CREATE_ID              TIMESTAMP (12) WITHOUT TIME ZONE
000012         GENERATED ALWAYS AS TRANSACTION START ID,
000013         PERIOD SYSTEM_TIME (SYS_STA, SYS_END),
000014         CONSTRAINT CUST_ID
000015         PRIMARY KEY (CUST_ID,
000016                    CATEGORY))
000017    PARTITION BY SIZE EVERY 4 G
000018    AUDIT NONE
000019    DATA CAPTURE NONE
000020    CCSID          EBCDIC
000021    NOT VOLATILE
000022    APPEND NO ;
000023    COMMIT;
***** ***** Bottom of Data *****

```

Figure 12-44 Generated DDL of CUSTOMER_COVERAGE table

Using Figure 12-45 as your guide, perform the following steps:

1. Change the table name from CUSTOMER_COVERAGE to HS_SUTOMER_COVERAGE at line 000002.
2. ADD a comma at the end of line 000007 after WITHOUT TIME ZONE NOT NULL for the SYS_STA column.
3. DELETE line 000008, which was the GENERATED ALWAYS AS ROW BEGIN attribute of SYS_STA column.
4. ADD a comma at the end of line 000009 after WITHOUT TIME ZONE NOT NULL for the SYS_END column.
5. DELETE line 000010, which was the GENERATED ALWAYS AS ROW END attribute of the SYS_END column.
6. ADD a comma at the end of line 000011 after WITHOUT TIME ZONE for the CREATE_ID column.
7. DELETE line 000012, which was the GENERATED ALWAYS AS TRANSACTION START ID attribute of the CREATE_ID column.
8. DELETE line 000013, which was the SYSTEM_TIEM period.

```

ISREDDE2   SYS10307.T154722.RA000.SYSADM.R0100184           Columns 00001 00072
Command ==>                                           Scroll ==> CSR
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001     SET CURRENT SQLID='ADMR2';
000002     CREATE TABLE ADMR2.HS_CUSTOMER_COVERAGE
000003         (CUST_ID                INTEGER NOT NULL,
000004          CATEGORY                CHAR(2) FOR SBCS DATA NOT NULL,
000005          DEDUCTABLE              DECIMAL(15, 2) NOT NULL,
000006          COVERAGE              DECIMAL(15, 2) NOT NULL,
000007          SYS_STA                TIMESTAMP (12) WITHOUT TIME ZONE NOT NULL,
000008          GENERATED ALWAYS AS ROW BEGIN,
000009          SYS_END                TIMESTAMP (12) WITHOUT TIME ZONE NOT NULL,
000010          GENERATED ALWAYS AS ROW END,
000011          CREATE_ID              TIMESTAMP (12) WITHOUT TIME ZONE,
000012          GENERATED ALWAYS AS TRANSACTION START ID,
000013          PERIOD SYSTEM_TIME (SYS_STA, SYS_END),
000014          CONSTRAINT CUST_ID
000015          PRIMARY KEY (CUST_ID,
000016                     CATEGORY))
000017     PARTITION BY SIZE EVERY 4 G
000014     AUDIT NONE
000015     DATA CAPTURE NONE
000016     CCSID          EBCDIC
000017     NOT VOLATILE
000018     APPEND NO ;
000019     COMMIT;
***** ***** Bottom of Data *****

```

Figure 12-45 Modifying the DDL of the CUSTOMER_COVERAGE table

Press Enter to update the DDL and then press PF3. The SQL - Execute & Copy panel (ADB2GEND) opens. Specify Yes to execute the generated SQL and press Enter (Figure 12-46).

```
ADB2GEND ----- VA1A SQL - Execute & Copy ----- 16:13
Option ==>
The SQL will be executed upon exiting this panel.
Execute the generated SQL : YES      (Yes/No)

Copy the SQL to dataset . :                               >
Disposition . . . . . :      (SHR, OLD or MOD)
The specified dataset will be created if it does not exist.
```

Figure 12-46 Execute the modified DDL

Press PF3. The DDL executes and the SET stmt executed message is displayed on the Tables, Views, and Aliases panel (ADB21T), which was the first statement in the modified DDL that created the HS_CUSTOMER_COVERAGE table.

12.4.6 ADD VERSIONING option

ADD VERSIONING is a DB2 Administration Tool V10 feature. It issues the ALTER TABLE ADD VERSIONING statement that is used with the USE HISTORY TABLE clause to define versioning on the table. By defining versioning, you establish a link between the system period temporal table and the history table.

Before the link is established, the history table is just a regular base table, with a table type of T (Figure 12-47). After the ADD VERSIONING statement is executed and the link between the system period temporal table and its corresponding history table is established successfully, the table type of the history table is changed to H.

```
ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols  Rows  ChksC
    *                  *      *  *      *      *      *    **
----->-----
      HS_CUSTOMER_COVERA ADMR2    T DSN00009 HSRCUSTO    7      -1    0
*****
***** END OF DB2 DATA *****
```

Figure 12-47 History table with table type T

To use the ADD VERSIONING option, specify s next to the ADD VERSIONING option on the Alter Table panel (ADB21TA) (Figure 12-48).

ADB21TA n ----- VA1A Alter Table -----

Command ==>

Table schema . . . : ADMR2 >

Table name . . . : CUSTOMER_COVERAGE >

AUDIT NONE (None, Changes, or All)

DATA CAPTURE NONE (None/Changes)

VALIDPROC NULL (NULL/Program name)

RESTRICT ON DROP . . . NO (Yes/No)

VOLATILE NO (Yes/No)

APPEND NO (Yes/No)

ALTER TABLE with any of the above changes OR select one of the options below

ADD column

ADD PRIMARY KEY

DROP PRIMARY KEY

ADD FOREIGN KEY

DROP FOREIGN KEY

ADD CHECK constraint

DROP CHECK constraint

ADD UNIQUE constraint

DROP UNIQUE constraint

ADD PERIOD

ACTIVATE ROW ACCESS CONTROL

DEACTIVATE ROW ACCESS CONTROL

ADD MATERIALIZED QUERY

DROP MATERIALIZED QUERY

REFRESH MATERIALIZED TABLE

ADD PARTITIONING KEY

ADD PARTITION

ADD CLONE

DROP CLONE

s ADD VERSIONING

DROP VERSIONING

ACTIVATE COLUMN ACCESS CONTROL

DEACTIVATE COLUMN ACCESS CONTROL

More: +

Figure 12-48 Select ADD VERSION option on ADB21TA

Press Enter. The Add Versioning panel (ADBPTAV) opens. You can either specify the name of the history table to be linked or use ? to look up the table name (Figure 12-49). Select the history table from the list of tables (Figure 12-50).

```
ADBPTAV n ----- VA1A Add Versioning -----
Command ==>

ALTER TABLE "ADMR2"."CUSTOMER_COVERAGE"
ADD VERSIONING USE HISTORY TABLE

Table schema . . ADMR2                (Optional, default is ADMR2)
Table name . . . ?                    (? to lookup)
```

Figure 12-49 Use look up to locate the history table

```
ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 5 of 5
Command ==>                                     Scroll ==> CSR
Select by typing '+'
Commands: GRANT MIG ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands
```

Sel	Name	Schema	T	DB Name	TS Name	Cols	Rows	ChksC
*	*	*	*	*	*	*	*	**
	NEWPHONE	DSN8A10	T	DSN8DA1U	NEWPHONE	7	-1	0
	ADBCPREREQ	SYSADM	T	ADBDCHA	ADBSCPRQ	7	-1	0
	CUSTOMER_COVERAGE	ADMR2	T	DSN00005	CUSTOMER	7	-1	0
	ADBHOLD	SYSADM	T	DADBA1A	ADBSHOLD	7	-1	0
+	HS_CUSTOMER_COVERA	ADMR2	T	DSN00009	HSRCUSTO	7	-1	0

```
***** END OF DB2 DATA *****
```

Figure 12-50 Select the history table from a list of tables

Press Enter. The history is selected and entered on the Add Versioning panel (ADBPTAV) (Figure 12-51).

```
ADBPTAV n ----- VA1A Add Versioning -----
Command ==>

ALTER TABLE "ADMR2"."CUSTOMER_COVERAGE"
ADD VERSIONING USE HISTORY TABLE

Table schema . . ADMR2                (Optional, default is ADMR2)
Table name . . . HS_CUSTOMER_COVERA  (? to lookup)
```

Figure 12-51 Specify the history table name

Press Enter. The ALTER stmt executed message is displayed on Alter Table panel (ADB21TA). The ADD VERSIONING statement was executed and the link between the HS_CUSTOMER_COVERAGE and the CUSTOMER_COVERAGE table is established successfully.

Now let us exam the type of the HS_CUSTOMER_COVERAGE table.

Press PF3 until you return to the DB2 System Catalog panel (ADB21). Enter HS_CUSTOMER_COVER at the name field and specify T at the Option line. Press Enter (Figure 12-52).

```

ADB21 min ----- VA1A System Catalog -----
Option ==> T

Object options:
AO - Authorization options
G - Storage groups
D - Databases
S - Table spaces
T - Tables, views, and aliases
V - Views
A - Aliases
Y - Synonyms
X - Indexes
C - Columns
N - Constraints
DS - Database structures
PDC - DB2 pending definition changes

P - Plans
L - Collections
K - Packages
M - DBRMs
H - Schemas
E - User defined data types
F - Functions
O - Stored procedures
J - Triggers
Q - Sequences
DSP - DS with plans and packages

More: +
DB2 System: VA1A
DB2 SQL ID: ADMR2

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name ==> HS_CUSTOMER_COVER > Grantor ==> >
Owner ==> > Grantee ==> >
In D/L/H ==> > Switch Catalog Copy ==> N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)

```

Figure 12-52 Locating the HS_CUSTOMER_COVERAGE table

Press Enter. The HS_CUSTOMER_COVERAGE table is displayed on the Tables, Views, and Aliases panel (ADB21T). Notice that the HS_CUSTOMER_COVERAGE table now has table type of H (Figure 12-53).

```

ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==>                                         Scroll ==> CSR

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols      Rows ChksC
    *                  *      *  *      *      *        *    **
----->-----
      HS_CUSTOMER_COVERA ADMR2    H DSN00009 HSRCUSTO    7        -1    0
*****
                                END OF DB2 DATA *****

```

Figure 12-53 History table with a table type of H

The SYSTEM_TIME period temporal table contains the system-maintained period in which DB2 maintains the beginning and ending timestamp values for a row. The old row that resulted from the DELETE and UPDATE operation is moved to the corresponding history table while the system_time bas temporal table contains the current rows.

In our example, there are give rows in the CUSTOMER_COVERAGE table (Figure 12-34 on page 334). We execute the following SQL statements and examine both the base temporal table and the history table afterwards:

- ▶ DELETE FROM ADMR2.CUSTOMER_COVERAGE WHERE CUST_ID = 333333;
- ▶ UPDATE ADMR2.CUSTOMER_COVERAGE SET DEDUCTABLE = 800.00 WHERE CUST_ID = 555555 AND CATEGORY = '02';
- ▶ INSERT INTO ADMR2.CUSTOMER_COVERAGE (CUST_ID, CATEGORY, DEDUCTABLE, COVERAGE) VALUES(555555, '01', 2000.00, 50000.00);
- ▶ COMMIT;

After these SQL statements successfully execute, we check the rows of data afterward in the table by issuing the L line command next to the CUSTOMER_COVERAGE table on the Tables, Views, and Aliases panel (ADB21T) and press Enter (Figure 12-54).

```
ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols  Rows ChksC
    *                  *      * *      *      *      *      *
-----
L    CUSTOMER_COVERAGE  ADMR2   T DSN00005 CUSTOMER    7      -1    0
*****
***** END OF DB2 DATA *****
```

Figure 12-54 Issue the L line command to show data rows

The CUSTOMER_COVERAGE base table shows that the row for CUST_ID 333333 is gone, the deductible amount for CUST_ID 555555's coverage catalog '02' has changed to 800.00, and a new entry named coverage category '01' has been added for CUST_ID 555555 (Figure 12-55).

```
ADB2BRIF -- VA1A BROWSE ADMR2.CUSTOMER_COV > ----- Line 00000000 Col 006 085
Command ==>                                     Scroll ==> CSR

***** Top of Data *****
CUST_ID CATEGORY          DEDUCTABLE COVERAGE SYS_STA
-----
111111 01                  500.00 10000.00 0001-01-01-00.00.00.0000000
222222 01                  500.00 20000.00 0001-01-01-00.00.00.0000000
444444 01                  1000.00 40000.00 0001-01-01-00.00.00.0000000
555555 02                  800.00 35000.00 2010-11-05-10.39.21.9104577
555555 01                  2000.00 50000.00 2010-11-05-10.39.21.9104577
***** Bottom of Data *****
```

Figure 12-55 CUSTOMER_COVERAGE system period temporal base table

Issue the L line command next to the HS_CUSTOMER_COVERAGE table. The deleted row for CUST_ID 333333 and the old row for CUST_ID 555555 before the UPDATE are now moved to the HS_CUSTOMER_COVERAGE, which is the history table of CUTOMER_COVERAGE (Figure 12-56).

```

ADB2BRIF  -- VA1A BROWSE ADMR2.HS_CUSTOMER_ >  ----- Line 00000000 Col 001
080
Command ==>                                     Scroll ==> CSR

***** Top of Data *****
CUST_ID CATEGORY          DEDUCTABLE COVERAGE SYS_STA
-----
333333 02                  1000.00 30000.00 0001-01-01-00.00.00.00
555555 02                  1000.00 35000.00 0001-01-01-00.00.00.00
***** Bottom of Data *****

```

Figure 12-56 HS_CUSTOMER_COVERAGE table contains archived rows

12.4.7 BASE line command

In addition to supporting the AUXiliary table and CLONE table, the BASE line command on the Tables, Views, and Aliases panel (ADB21T) has been enhanced in DB2 Administration Tool V10 to support the history table. Enter the BASE line command next to the HS_CUSTOMER_COVERAGE that is next to the history table (Figure 12-57).

```

ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel   Name                Schema  T DB Name  TS Name   Cols      Rows ChksC
*      *                  *      *  *      *
----->-----
base  HS_CUSTOMER_COVERA  ADMR2   H DSN00009 HSRCUSTO    7        -1    0
***** END OF DB2 DATA *****

```

Figure 12-57 Enhanced BASE line command

Press Enter. The corresponding base temporal table, the CUSTOMER_COVERAGE table with type T, opens (Figure 12-58).

```
ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols      Rows ChksC
    *                  *      *  *      *      *        *    **
-----
      CUSTOMER_COVERAGE  ADMR2   T DSN00005 CUSTOMER    7        -1    0
*****
                                END OF DB2 DATA *****
```

Figure 12-58 Base temporal table

Press PF3 until you return to the Tables, View, and Aliases panel (ADB21T) and issue the ALT line command against the HS_CUSTOMER_COVERAGE table. The Invalid object type message is displayed under the command line, which indicates that you cannot use ALT on a table that has a table type of H - History table (Figure 12-59).

```
ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR
Invalid object type
Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols      Rows ChksC
    *                  *      *  *      *      *        *    **
----->-----
ALT  HS_CUSTOMER_COVERA  ADMR2   H DSN00009 HSRCUSTO    7        -1    0
*****
                                END OF DB2 DATA *****
```

Figure 12-59 ALT command not allowed for history table

The ALT command does not support the versioned table. To use ALT on a temporal table or history table, you need to drop the link between the system-maintained temporal table and its associated history table.

12.4.8 DROP VERSIONING option

DROP VERSIONING is a DB2 Administration Tool V10 feature. It issues the ALTER TABLE DROP VERSIONING statement (Figure 12-60). The definition of the columns and data of the table are not changed, but the table is no longer treated as a system-maintained temporal table. The SYSTEM_TIME period is retained and the history table is not dropped. Only the relationship between the system-maintained temporal table and the history table is removed. (Figure 12-60).

```
ADB21TA n ----- VA1A Alter Table -----
Command ==>

Table schema . . : ADMR2      >
Table name . . . : CUSTOMER_COVERAGE  >

AUDIT . . . . . NONE      (None, Changes, or All)
DATA CAPTURE . . . . . NONE (None/Changes)
VALIDPROC . . . . . NULL   (NULL/Program name)
RESTRICT ON DROP . . . NO   (Yes/No)
VOLATILE . . . . . NO      (Yes/No)
APPEND . . . . . NO        (Yes/No)

ALTER TABLE with any of the above changes OR select one of the options below

                                     More:      +
ADD column                          ADD MATERIALIZED QUERY
ADD PRIMARY KEY                     DROP MATERIALIZED QUERY
DROP PRIMARY KEY                    REFRESH MATERIALIZED TABLE
ADD FOREIGN KEY                     ADD PARTITIONING KEY
DROP FOREIGN KEY                    ADD PARTITION
ADD CHECK constraint                ADD CLONE
DROP CHECK constraint                DROP CLONE
ADD UNIQUE constraint                ADD VERSIONING
DROP UNIQUE constraint              S DROP VERSIONING
ADD PERIOD                          ACTIVATE COLUMN ACCESS CONTROL
ACTIVATE ROW ACCESS CONTROL          DEACTIVATE COLUMN ACCESS CONTROL
DEACTIVATE ROW ACCESS CONTROL
```

Figure 12-60 DROP VERSIONING option

After the versioning is dropped, the table type of the HS_CUSTOMER_COVERAGE table is changed from H to T (Figure 12-61).

```
ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 1 of 1
Command ==>                                     Scroll ==> CSR

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name  TS Name  Cols  Rows ChksC
    *                  *      *  *      *      *      *   **
----->-----
      HS_CUSTOMER_COVERA ADMR2    T DSN00009 HSRCUSTO    7      -1    0
*****
                                END OF DB2 DATA *****
```

Figure 12-61 HS_CUSTOMER_COVERAGE with a table type of T

The contents of the history table remain and you can now drop the history table if you choose to do so.



Security

In this chapter, we look at the new security features introduced in DB2 10 for z/OS to help customers meet the challenges of compliance and securing their data, and how DB2 Administration Tool supports these features. We also look at security support delivered in DB2 9 for z/OS.

This chapter discusses the following topics:

- ▶ Overview of security features
- ▶ Separation of duties
- ▶ Revoking authorities
- ▶ Copying authorities
- ▶ Defining security constructs
- ▶ Auditing profiles

13.1 Overview of security features

A major feature of compliance is the role of trusted users. These are users who have access to data because of their security level and not because of an underlying business need. There is now a greater need to monitor the activities of these users and to report on their activity. DB2 10 for z/OS delivers mechanisms to audit, restrict, and separate these two compliance issues, namely, separation of duty and restriction of data access.

DB2 10 for z/OS provides several new features that help companies meet these challenges, namely the ability of users to set audit policies to identify system activity, separation of duties, and row and column access controls.

We look in detail at how DB2 Administration Tool assists DBAs in creating and maintaining these features and at how you can manage Trusted Contexts and Roles that were delivered with DB2 9 for z/OS. We do not deal with the usage of these authorities; that topic is covered in other IBM Redbooks publications.

13.2 Separation of duties

There are new levels of granularity of security within DB2 10 for z/OS to allow for separation of duties. These authorities allow the separation of security tasks from system tasks, protect sensitive data from privileged users, and separate SQL tuning from data access. These new authorities are shown in Figure 13-1, along with authorities that existed prior to DB2 10 for z/OS.

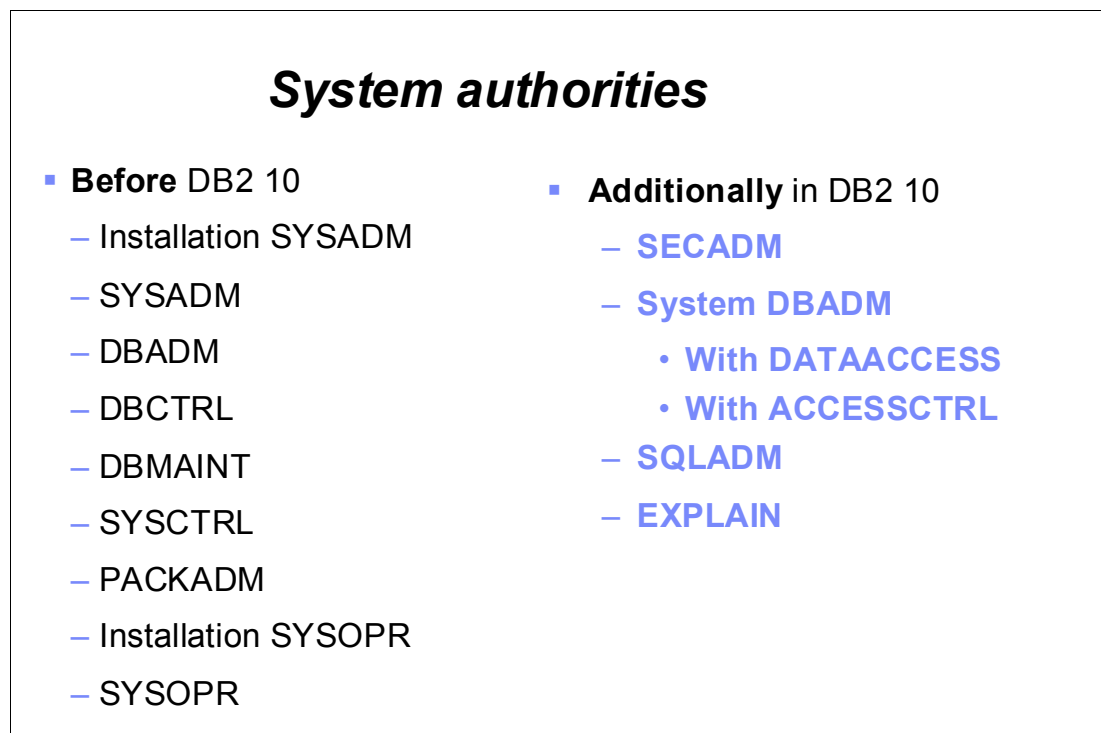


Figure 13-1 DB2 system authorities

We look at how these new authorities are granted and displayed within the DB2 Administration Tool. A full description of these authorities is defined and discussed in detail in *DB2 10 for z/OS Technical Overview*, SG24-7892.

A brief summary of the new authorities follows:

- ▶ SECADM
 - Performs security related tasks with no inherent data access.
- ▶ System DBADM
 - Allows DB2 object management in the subsystem.
 - With or without access to data.
 - With or without control over access to data.
- ▶ SQLADM includes
 - Issues SQL EXPLAIN statements.
 - Start, stops, and displays profiles.
 - Execute RUNSTATS and modify statistics.
 - EXPLAIN privilege.
 - Cannot access data, run DDL, or execute.
- ▶ EXPLAIN
 - Issues an SQL EXPLAIN PLAN/ALL statement without requiring data privilege.
 - Issues SQL PREPARE or DESCRIBE without privilege.
 - Can specify BIND EXPLAIN (ONLY) and SQLError(CHECK).
 - Can explain dynamic SQL using a special register.

SECADM is a DSNZPARM set user ID and is included for completeness, but is not maintainable outside of DB2 Security within the DSNZPARM. SECADM is not discussed further, except to say that it is updateable by using the DSNZPARM panels in DB2 Administration Tool. You can use option Z.2Z to display the DSNZPARMS and, as you can see from Figure 13-2, the SECADM value is online updateable.

```

ADB2ZZMN ----- VA1A System Parameters - System Parameters -----14:54
Command ==>

                                DB2 System: VA1A
                                DB2 SQL ID: SYSADM
                                More:      - +
(*) Online changeable parameter

Wait for retained locks . . . . . 1 (RETLWAIT) *
Enables data retrieval from index key . . . . . NO (RETVLCFK) *
Revoke dependent privileges . . . . . SQLSTMT (REVOKE_DEP_PRIV) *
Resource limit table creator . . . . . SYSADM (RLFAUTH) *
Resource limit specification error action . . . . . 12240 (RLFERR) *
DDF RLF error parameter . . . . . NOLIMIT (RLFERRD) *
Resource limit specification table suffix. . . . . 01 (RLFTBL) *
Reordered row format . . . . . ENABLE (RRF) *
U Lock for RR or RS . . . . . YES (RRULOCK) *
Security administrator 1 . . . . . SECADM > (SECADM1) *
Security administrator 1 input style . . . . . CHAR (SECADM1_INPUT_) *
Security administrator 1 type . . . . . AUTHID (SECADM1_TYPE) *
Security administrator 2 . . . . . SECADM > (SECADM2 ..) *
Security administrator 2 input style . . . . . CHAR (SECADM2_INPUT_) *
Security administrator 2 type . . . . . AUTHID (SECADM2_TYPE) *
Secondary space allocation. . . . . 128 (SECQTY) *
Separate security between SYSADM/SECADM . . . . . NO (SEPARATE_SECUR) *

```

Figure 13-2 Updating DSNZPARM SECADM user ID

All of the new authorities are system authorities and are all controlled by the ZA - System authorizations option of the System Catalog panel (AO option) (Figure 13-3). The System DBADM is a system authority, and is not to be confused with the database DBADM authority. It is also worthwhile to discuss the new options of the System DBADM, DATAACCESS, and ACCESSCTRL. These features allow you to grant System DBADM without giving access to the data in all user tables, views, and so on (DATAACCESS) or the ability to grant and revoke privileges on all resources in a DB2 subsystem (ACCESSCTRL).

```
DB2 Admin ----- VA1A System Catalog ----- 14:14
Option ==>

                                     More:  +
Authorization options:                DB2 System: VA1A
    OO - Object options                DB2 SQL ID: SYSADM
    GA - Storage group auths          PA - Plan authorizations
    DA - Database authorizations      LA - Collection authorizations
    SA - Table space authorizations   KA - Package authorizations
    TA - Table authorizations         HA - Schema authorizations
    VA - View authorizations          EA - User defined data type authorizations
    CA - Column authorizations        FA - Function authorizations
    ZA - System authorizations       OA - Stored procedure authorizations
    UA - User authorizations          QA - Sequence authorizations
    RA - Resource authorizations      TR - Trusted contexts
    RO - Roles                       PM - Permissions
    CM - Column masks

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name      ==>                > Grantor ==>                >
Owner     ==>                > Grantee ==>                >
In D/L/H ==>                > Switch Catalog Copy ==> N   (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
```

Figure 13-3 Authorization options

After selecting ZA, a panel detailing the system authorities that are currently granted in the DB2 Subsystem opens. Issue the GR line command (Figure 13-4).

```
ADB2AZ in ----- VA1A System Privileges Authorizations --- Row 1 to 4 of 4
Command ==>                                     Scroll ==> CSR

Commands: REVOKE  GRANT  SYSAUTH
Line commands:
R - Revoke GR - Grant          B B  CREATE   : S B M M S S S D E S S D A
I - Interpretation            I S              S T I O O Y Y Y E X Q Y A C
RE - Grantee role             N D              A T E O N N N S S S B P L D T C
RR - Grantor role             D S              L M C S D 1 2 A C O U L A B A E
                                A  D D  I T U P A      D T P G A D A A S
                                H D  B B S A A R A G      M R R E I M D C S
Sel Grantor  Grantee  T Grant date  G D  A C G S B E C T      L  S N  M C C
*           *        * *          * * * * * * * * * * * * * * * * * *
-----
      SYSIBM  SYSOPR    1985-04-01      G
      SYSADM  NEWTON    2009-04-23  S      Y
      SYSADM  SYSOPR    2009-04-30  S      Y
gr  SYSADM  ADMF001    2009-04-30  S      Y
***** END OF DB2 DATA *****
```

Figure 13-4 System Privileges Authorizations panel

The Grant System Privileges panel opens, where you can build your GRANT statement (Figure 13-5).

```

ADB2GZ in ----- VA1A Grant System Privileges -----
15:03
Command ==>

GRANT                                                                    More:      +

Specify Y or G (for with grant option) or ' ' (for none)

SYSADM          BSDS          CREATESG          STOPALL
SYSOPR          CREATEDBA     DISPLAY          STOSPACE
BINDADD         CREATEDBC     RECOVER          TRACE
MONITOR1        MONITOR2     CREATEALIAS      SYSCTRL
BINDAGENT       ARCHIVE       CREATETMTAB     DEBUGSESSION
Y EXPLAIN       Y SQLADM      Y ACCESSCTRL    Y DATAACCESS
Y CREATE_SECURE_OBJECT  Y DBADM

Yes/No is only valid below when DBADM is specified with Y above.

WITH ACCESSCTRL . . YES  (Yes/No)
WITH DATAACCESS . . YES  (Yes/No)

T0

To . . sg247916

```

Figure 13-5 Grant System Privileges panel

In this example, we are granting every new authority (without the GRANT option) possible, with System DBADM receiving both DATAACCESS and ACCESSCTRL authority, to show how to grant all the new authorities. Figure 13-6 shows the result of building the grant statements (this session has the prompts turned on to allow for screen captures).

```
ADB2PSTM ----- VA1A Statement Execution Prompt ----- 15:07
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'SYSADM.AAEKHS.WSL'
Work statement list name ==> R0000001 Action ==> A (Append or Replace)
More: +

Statement that is about to be executed (first 28 lines):
GRANT EXPLAIN,SQLADM,ACCESSCTRL,DATAACCESS,CREATE_SECURE_OBJECT,DBADM
TO "SG247916"
```

Figure 13-6 Built GRANT statement

Press Enter. You return to the refreshed list of system authorization panel (Figure 13-7).

```
ADB2AZ in ----- VA1A System Privileges Authorizations --- Row 1 to 5 of 5
Command ==> Scroll ==> CSR

Commands: REVOKE GRANT SYSAUTH
Line commands:
R - Revoke GR - Grant          B B CREATE : S B M M S S S D E S S D A
I - Interpretation             I S          S T I O O Y Y Y E X Q Y A C
RE - Grantee role              N D          A T E O N N N S S S B P L D T C
RR - Grantor role              D S          L M C S D 1 2 A C O U L A B A E
                                A D D I T U P A          D T P G A D A A S
                                H D B B S A A R A G          M R R E I M D C S
Sel Grantor Grantee T Grant date G D A C G S B E C T          L S N M C C
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
-----
SYSIBM SYSOPR 1985-04-01 G
SYSADM NEWTON 2009-04-23 S Y
SYSADM SYSOPR 2009-04-30 S Y
SYSADM ADMF001 2009-04-30 S Y
* SYSADM SG247916 2010-10-20 S Y Y Y Y Y
***** END OF DB2 DATA *****
```

Figure 13-7 Updated system authorizations

13.3 Revoking authorities

To revoke an authority, issue R against the authority you want to revoke. In our example, we revoke the ability to create secured objects from SG247916. (Figure 13-8).

```
ADB2RZ in ----- VA1A Revoke System Privileges ----- 15:27
Command ==>

REVOKE                                     DB2 SQL ID: SYSADM

Enter any character in front of the privilege to revoke it from the user:

  SYSADM      BSDS      CREATESG      STOPALL
  SYSOPR      CREATEDBA  DISPLAY      STOSPACE
  BINDADD     CREATEDBC  RECOVER     TRACE
  MONITOR1    MONITOR2   CREATEALIAS  SYSCTRL
  BINDAGENT   ARCHIVE    CREATETMTAB  DEBUGSESSION
Y EXPLAIN     Y SQLADM     Y DBADM      Y DATAACCESS
Y ACCESSCTRL  Y CREATE_SECURE_OBJECT

FROM
  From . . . . . SG247916 >
BY
  By . . . . . >
INCLUDING DEPENDENT PRIVILEGES
  Cascade revoke . . YES (Yes/No)

Report Revoke Impacts . . . YES (Yes/No)
Report Dropped Synonyms & Aliases . . NO (Yes/No)
```

Figure 13-8 Revoking system authorities

Remove the Y from the authorities that you want to remain authorized. It is worth reviewing the Revoke Impact report before carrying out the revoke (Figure 13-9).

```
ADB2RIP n ----- VA1A Revoke Impact Report ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> PAGE

Line commands: I - Interpretation
                                Owner/
S   Grantee  G Resource N/  O Schema/  Grantor/  G H Privileges/
Lv      T Collection  T P/K Name Binder  T G Effect
-----
* 0 SG247916          Z          SYSADM          Y
***** END OF DB2 DATA *****
```

Figure 13-9 Revoke Impact Report panel

The OT column has a one character code representing the type of authority that will be revoked. The codes are shown in Table 13-1.

Table 13-1 Object code

Object code	Object description
G	Storage group
D	Database
S	Table space
T	Table
P	Plan
K	Package
L	Collection
E	Distinct type
B	Buffer pool
Z	System
H	Schema
F	User defined function
O	Stored procedure

Return to the panel shown in Figure 13-8 on page 365, change the Report Revoke Impact to NO, and run the REVOKE command. You receive a message informing you of the outcome of the command.

13.3.1 Revoking authorities without cascading

A new function delivered with DB2 10 for z/OS is the REVOKE....DEPENDENT clause. This clause allows you to decide whether you want the Revoke statement to cascade through the system and remove all dependent authorities. DB2 Administration Tool V10 supports this clause through the Revoke Table Privileges panel (ADB2RT). From the Table Authorizations panel, enter R against the authority that you want to revoke (Figure 13-10).

ADB2AT in ----- DB0B Table Authorizations -----										Row 1 to 2 of 2									
Command ==>										Scroll ==> PAGE									
Commands: REVOKE GRANT																			
Line commands:																			
R - Revoke GR - Grant T - Table I - Interpretation										U D I S U R									
CA - Column authorizations RE - Grantee role										P A E I N E P R E									
RR - Grantor role										D L L N S L D E F T									
										C T E D E E A F C R									
										O E T E R C T E O I									
S	Grantor	Grantee	G	T Schema	Name	H Date	G Grant	L R E X T T E R L G											
	*	*	* *	*	*	* *	*	* * * * * * * *											

	ADMR3	ADMR3		ADMR3	EMP	101008		G G G G G G G	G										
r	ADMR3	ADMR2		ADMR3	EMP	101026		G G G G G G G	G										
***** END OF DB2 DATA *****																			

Figure 13-10 Revoke table authorities

The panel shown in Figure 13-11 appears. Note the highlighted field, which is where you select to revoke dependent authorities or not.

```

ADB2RT in ----- DB0B Revoke Table Privileges ----- 17:17
Command ==>

REVOKE                                     DB2 SQL ID: ADMR3

Enter any character in front of the privilege to revoke it from the user:

  ALL          G INDEX          G UPDATE
G ALTER        G INSERT        G REFERENCE
G DELETE       G SELECT        G TRIGGERS

ON TABLE
  Owner . . . ADMR3    >
  Table . . . EMP      >
FROM
  From . . . . ADMR2    >
BY
  By . . . . .         >

INCLUDING DEPENDENT PRIVILEGES
  Cascade revoke . . . (Yes/No)

Report Revoke Impacts . . . YES (Yes/No)
Report Dropped Synonyms & Aliases . . NO (Yes/No)

```

Figure 13-11 Revoke dependent authorities

If you select YES, you generate the statement shown in Example 13-1, which gives the same results as in prior DB2 versions.

Example 13-1 Generated Revoke Dependent statement

```

REVOKE ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER
  ON TABLE "ADMR3"."EMP"
  FROM "ADMR2"
  INCLUDING DEPENDENT PRIVILEGES

```

If you select NO, you generate the statement shown in Example 13-2.

Example 13-2 Generate Revoke Not Dependent statement

```

REVOKE ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER
  ON TABLE "ADMR3"."EMP"
  FROM "ADMR2"
  NOT INCLUDING DEPENDENT PRIVILEGES

```

13.4 Copying authorities

DB2 Administration Tool provides you with a method of copying authorities from one user to another user. which is useful when setting up new environments or creating new users.

The example we are going to use is to copy all the table authorizations from DSN8A10 to ADMR3. To start the process, we select Table authorizations from the System Catalog panel (Figure 13-12).

```
ADB21 min ----- VA1A System Catalog ----- 15:12
Option ==> TA

                                     More:      +
Authorization options:                DB2 System: VA1A
  OO - Object options                  DB2 SQL ID: SYSADM
  GA - Storage group auths             PA - Plan authorizations
  DA - Database authorizations         LA - Collection authorizations
  SA - Table space authorizations      KA - Package authorizations
  TA - Table authorizations            HA - Schema authorizations
  VA - View authorizations             EA - User defined data type authorizations
  CA - Column authorizations           FA - Function authorizations
  ZA - System authorizations           OA - Stored procedure authorizations
  UA - User authorizations             QA - Sequence authorizations
  RA - Resource authorizations         TR - Trusted contexts
  RO - Roles                          PM - Permissions
  CM - Column masks

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name      ==>                > Grantor ==>                >
Owner      ==>                > Grantee ==>                >
In D/L/H ==>                > Switch Catalog Copy ==> N   (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
```

Figure 13-12 Selecting table authorizations

We could include the Grantee on this panel to narrow down the results set for the next panel, or we can use the SEARCH facility (Figure 13-13).

ADB2AT in ----- VA1A Table Authorizations ----- Row 6 from 737
Command ==> Scroll ==> CSR

Commands: REVOKE GRANT
Line commands:
R - Revoke GR - Grant T - Table I - Interpretation U D I S U R
CA - Column authorizations RE - Grantee role P A E I N E P R E
RR - Grantor role D L L N S L D E F T
C T E D E E A F C R
O E T E R C T E O I
L R E X T T E R L G
* * * * * * * * * *

S	Grantor	Grantee	G T Schema	Name	H Date G Grant										
	*	DSN8A10*	* *	*	* *	* * * * * * * * * *									
---	---	---	---	---	---	---									
	DSN8A10	DSN8A10	DSN8A10	DSN_QUERY_AUX	S 090424	G	G	G	G	G	G	G	G	G	
	DSN8A10	DSN8A10	DSN8A10	DSN_STATEMENT_CAC	S 090424	G	G	G	G	G	G	G	G	G	
	DSN8A10	DSN8A10	DSN8A10	VEMPDPT1	090423								G		
	DSN8A10	DSN8A10	DSN8A10	VEMPDPT1	090423								G		
	DSN8A10	DSN8A10	DSN8A10	VDEPMG1	090423								G		
	DSN8A10	DSN8A10	DSN8A10	VDEPMG1	090423								G		
	DSN8A10	DSN8A10	DSN8A10	VHDEPT	090423					G	G	G	G		
	DSN8A10	DSN8A10	DSN8A10	VDEPT	090423					G	G	G	G		
	DSN8A10	DSN8A10	DSN8A10	DEPT	S 090423	G	G	G	G	G	G	G	G	G	
	DSN8A10	DSN8A10	DSN8A10	VEMPLP	090423					G			G	G	
	DSN8A10	DSN8A10	DSN8A10	VPHONE	090423								G		

Figure 13-13 Table authorizations for DSN8A10

Note: To copy the authorizations, you can only have one Grantee on this panel. You cannot copy authorizations from more than one Grantee in one step.

From this panel, we issue the Grant statement for all of the statements in the list. You are prompted for the grantees (Figure 13-14). If you are using multiple grantees, then ensure you separate each one by a comma.

ADB2RALL ----- Grant -----

Specify grantees to use for all the GRANT statements:

Grantees . . ADMR3

Figure 13-14 Entering grantees

After entering the grantees, the grants are either run, or the panel shown in Figure 13-15 opens, depending upon your prompt options.

```
ADB2PSTM ----- VA1A Statement Execution Prompt ----- 15:35
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now (Add an A for all stmts. For example 1A - Execute all stmts):
    1 - Execute the statement
    2 - Edit the statement
    3 - Create a batch job with the statement
    4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'SYSADM.AAEKHS.WSL'
Work statement list name ==> R0000001 Action ==> A (Append or Replace)
More: +
Statement that is about to be executed (first 28 lines, more stmts pending):
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER ON TABL
E "DSN8A10"."ACT" TO ADMR3 WITH GRANT OPTION
```

Figure 13-15 Statement Execution Prompt

You will be prompted for each individual statement unless you use the A suffix, which acts upon all statements. If you want to run all the statements, use the 1A command. We want to run it in batch to show all the statements, so issue 3A; if you do not use the A option, you create a batch job for only the statement shown on the panel. Option 3A generates the job shown in Example 13-3 (truncated for space reasons).

Example 13-3 Copying authorizations batch job

```
//SYSADMD JOB (SETUP),'TESTCASE',
//*      RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
//      REGION=OM,NOTIFY=SYSADM,
//      MSGCLASS=7
.....
//SYSTSPRT DD SYSOUT=*
//ADBRPTSM DD SYSOUT=*
//SYSTSIN DD *
    DSN SYSTEM(VA1A)
    RUN PROGRAM(ADBTEP2) PLAN(ADBTEP2) -
        PARMS('/SSID(VA1A) WORKLIST(DB2BATCH) -
    RESTART(NO) -
    ')
    END
//ADBTPEIN DD *
BINDERROR='MAXE'
;
//SYSIN DD *
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER ON
TABLE "DSN8A10"."ACT" TO ADMR3 WITH GRANT OPTION;
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER ON
TABLE "DSN8A10"."AGEGROUP" TO ADMR3 WITH GRANT OPTION;
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES ON TABLE
"DSN8A10"."AUX_BMP_PHOTO" TO ADMR3 WITH GRANT OPTION;
```

```

GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES ON TABLE
"DSN8A10"."AUX_EMP_RESUME" TO ADMR3 WITH GRANT OPTION;
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES ON TABLE
"DSN8A10"."AUX_PSEG_PHOTO" TO ADMR3 WITH GRANT OPTION;
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER ON
TABLE "DSN8A10"."CATALOG" TO ADMR3 WITH GRANT OPTION;
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER ON
TABLE "DSN8A10"."CITY" TO ADMR3 WITH GRANT OPTION;
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER ON
TABLE "DSN8A10"."CUSTOMER" TO ADMR3 WITH GRANT OPTION;
.....
GRANT SELECT ON TABLE "DSN8A10"."MATPROD" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MATPROD" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MATPROD" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MATPROD" TO ADMR3 WITH GRANT OPTION;
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER ON
TABLE "DSN8A10"."MQT1320" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT1320" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT1320" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT1320" TO ADMR3 WITH GRANT OPTION;
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER ON
TABLE "DSN8A10"."MQT2222" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT2222" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT2222" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT2222" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT2222" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT2222" TO ADMR3 WITH GRANT OPTION;
GRANT ALTER,DELETE,INDEX,INSERT,SELECT,UPDATE,REFERENCES,TRIGGER ON
TABLE "DSN8A10"."MQT2331" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT2331" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT2331" TO ADMR3 WITH GRANT OPTION;
GRANT SELECT ON TABLE "DSN8A10"."MQT2331" TO ADMR3 WITH GRANT OPTION;
.....

```

After the job is submitted, the copy process is complete. This process can be used for all authorizations.

13.5 Defining security constructs

In this section, we look at how to define column masks, row permissions, trusted contexts, and roles using the DB2 Administration Tool. For a full description of these security constructs, see Chapter 10, “Security”, in *DB2 10 for z/OS Technical Overview*, SG24-7892.

Here is a brief description of each construct:

- **Column masks**

Column masks allow you to define how a column value is returned to the user. Session information can be used to mask data, for example, salary can only be seen by users in certain groups; otherwise a NULL is returned.

- Row permissions

Controls who can see what rows in a table. It is applicable to SELECT, INSERT, UPDATE, DELETE, and MERGE. Permissions can also use session variable to control what rows are returned.

- Trusted context

Trusted contexts establishes a trusted relationship between DB2 and an external entity, such as a user ID, server ID, or IP address. Once created, a trusted context can use privileges held by a role.

- Roles

An entity that groups privileges together and can be assigned to a user ID or trusted audit policies.

13.5.1 Columns masks

Column masks are constructs that are defined at a table level that use SQL to control access to a table at the column level. A column mask is like a scalar function that returns a masked value for a column. DB2 Administration Tool adds value to the management of this security construct by providing easy, panel-driven methods of creating, viewing, and interpreting the masks. To use a column mask, you must have SECADM authority, create the mask, and activate the mask at the table level. This activation immediately starts the masking process. We look at how to define, activate, view, and deactivate column masks.

Creating a column mask

To create a column mask, you can use the DDL command (option 2.4), which opens the Create/Drop/Label/Comment on Objects panel (Figure 13-16).

```
ADB26 min ----- VA1A Create/Drop/Label/Comment On Objects ----- 08:34
Option ==>
```

		More: +
		DB2 System: VA1A
		DB2 SQL ID: SYSADM
CREATE	DROP	
CG - Storage group	DG - Storage group	
CD - Database	DD - Database	
CS - Table space	DS - Table space	
CT - Table	DT - Table	
CV - View	DV - View	
CL - Alias	DL - Alias	
CX - Index	DX - Index	
CY - Synonym	DY - Synonym	
CA - Auxiliary table		
CE - Distinct type	DE - Distinct type	
CJ - Trigger	DJ - Trigger	
CF - Function	DF - Function	
CO - Stored procedure	DO - Stored procedure	
CM - Materialized table		
CQ - Sequence	DQ - Sequence	
CTR - Trusted context	DTR - Trusted context	
CRO - Role	DRO - Role	
CCM - Column mask	DCM - Column mask	

Figure 13-16 Create/Drop/Label/Comment on Objects panel

Select the CCM (Create Column Mask) option. The Create Column Mask panel opens (Figure 13-17). Alternatively, you can get to this panel by selecting CRE from the Column Mask List panel.

```
ADBP6CCM ----- VA1A Create Column Mask ----- 08:37
Command ==>

Commands: EDIT  COPY  CREATE

CREATE MASK
Schema . . . . . > (default is SYSADM)
Name . . . . . > (? to look up)
ON (Table)
Schema . . . . . > (default is SYSADM)
Name . . . . . > (? to look up)
AS (Correlation)
Name . . . . . >
FOR COLUMN
Name . . . . . (? to look up)
RETURN (Expression): (first 5 lines, use EDIT for all lines)
CASE

END
ENABLE/DISABLE
Initial state . . DISABLE (Enable/Disable)
```

Figure 13-17 Initial Create Column Mask panel

Provide the details of the mask you want to create. If you need to check any of details of the table or the column on which you want to define the mask, use the “?” lookup command and select the correct values. In our example, we create the mask that is referenced in *DB2 10 for z/OS Technical Overview*, SG24-7892.

First, enter the details shown in Figure 13-18.

```
ADBP6CCM ----- VA1A Create Column Mask ----- 09:08
Command ==>

Commands: EDIT  COPY  CREATE

CREATE MASK
Schema . . . . . ADMR3      >      (default is SYSADM)
Name   . . . . . INCOME_BRANCH >    (? to look up)
ON (Table)
Schema . . . . . ADMR3      >      (default is SYSADM)
Name   . . . . . CUSTOMER    >    (? to look up)
AS (Correlation)
Name   . . . . .             >
FOR COLUMN
Name   . . . . . INCOME      (? to look up)
RETURN (Expression):      (first 5 lines, use EDIT for all lines)
    CASE
    END

ENABLE/DISABLE
Initial state . . DISABLE      (Enable/Disable)
```

Figure 13-18 Create Column Mask panel

To enter the CASE expression, select the EDIT command. An ISPF edit session where you can enter the details opens (Example 13-4).

Example 13-4 Entering Column Mask Expressions

```
000001 CASE
000002   WHEN
000003     (VERIFY_GROUP_FOR_USER(SESSION_USER, 'DBOB#A', 'DBOB#B', 'DBOB#C'
000004     ) = 1 AND (SUBSTR(CURRENT SQLID, 6, 1) = BRANCH)) THEN
000005       ADMR3.CUSTOMER.INCOME
000006   ELSE
000007     NULL
000008 END
```

When you return to the Create Column Mask panel, you see that the RETURN (Expression) is completed and that the details are entered in the edit session (Figure 13-19).

```
ADBP6CCM ----- VA1A Create Column Mask ----- 10:28
Command ==>

Commands: EDIT  COPY  CREATE

CREATE MASK
Schema . . . . . ADMR3      >      (default is SYSADM)
Name . . . . . INCOME_BRANCH >      (? to look up)
ON (Table)
Schema . . . . . ADMR3      >      (default is SYSADM)
Name . . . . . CUSTOMER     >      (? to look up)
AS (Correlation)
Name . . . . .              >
FOR COLUMN
Name . . . . . INCOME       (? to look up)
RETURN (Expression):      (first 5 lines, use EDIT for all lines)
    CASE
    WHEN
        (VERIFY_GROUP_FOR_USER(SESSION_USER, 'DBOB#A', 'DBOB#B', 'DBOB#C'
        ) = 1 AND (SUBSTR(CURRENT SQLID, 6, 1) = BRANCH)) THEN
        ADMR3.CUSTOMER.INCOME
ENABLE/DISABLE
Initial state . . DISABLE      (Enable/Disable)
```

Figure 13-19 Create Column Mask panel with RETURN (Expression)

We can now create the mask by selecting CREATE. This option either runs the command directly or takes you to the Statement Execution Prompt panel (Figure 13-20).

```
ADB2PSTM ----- VA1A Statement Execution Prompt ----- 10:29
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'SYSADM.WSL.JCL'
Work statement list name ==> Action ==> A (Append or Replace)
More: +

Statement that is about to be executed (first 28 lines):
CREATE MASK
  "ADMR3"."INCOME_BRANCH"
ON "ADMR3"."CUSTOMER"

FOR COLUMN INCOME RETURN
CASE
  WHEN
    (VERIFY_GROUP_FOR_USER(SESSION_USER, 'DBOB#A', 'DBOB#B', 'DBOB#C'
    ) = 1 AND (SUBSTR(CURRENT SQLID, 6, 1) = BRANCH)) THEN
    ADMR3.CUSTOMER.INCOME
  ELSE
    NULL
END
DISABLE
```

Figure 13-20 Statement Execution Prompt panel

By selecting Option 2, you can see the statement that is about to be executed (Example 13-5).

Example 13-5 Generated Create Mask statement

```
000001 CREATE MASK
000002   "ADMR3"."INCOME_BRANCH"
000003   ON "ADMR3"."CUSTOMER"
000004   FOR COLUMN INCOME RETURN
000005   CASE
000006     WHEN
000007       (VERIFY_GROUP_FOR_USER(SESSION_USER, 'DBOB#A', 'DBOB#B', 'DBOB#C'
000008       ) = 1 AND (SUBSTR(CURRENT SQLID, 6, 1) = BRANCH)) THEN
000009         ADMR3.CUSTOMER.INCOME
000010     ELSE
000011       NULL
000012   END
000013   DISABLE
```

Note that this mask is currently disabled. See “Activating column masks” on page 379 for details about activating the column mask.

When creating or activating column masks or row permissions, you have to consider any triggers defined on the tables. A trigger is usually used to enforce database integrity and therefore must have access to data that has not been filtered by column or row access control rules. If you limit access for the trigger, then the integrity of the database may be compromised. If you have a trigger on a table where you want to add row or column access controls, then you *must* alter or define the trigger as SECURE. To alter the trigger, issue the AL(ter) line command against the trigger. The Alter Trigger panel opens (Figure 13-21), where you can alter the trigger definition.

```
ADB21JA n ----- VA1A Alter Trigger ----- 16:11
Command ==>

ALTER TRIGGER

Schema . . . . . ADMR3                (Optional)
Name . . . . . CONNECTINFOCONSTRA >   (? to look up)

SECURED . . . . . YES                (Yes/No)
```

Figure 13-21 Altering a Trigger to SECURE

If you do not use a secure trigger, then it will be subject to access controls.

Activating column masks

After a column mask is defined, you activate it by altering the table (you can activate it before defining the mask, if required). Activate the column mask by issuing the AL(ter) command against the table that has the column mask. After the AL is issued, the Alter Table panel opens (Figure 13-22).

DB2 Admin ----- VA1A Alter Table -----14:54

Command ==>

Table schema . . . : ADMR2 >

Table name . . . : CUSTOMER_COVERAGE >

AUDIT NONE (None, Changes, or All)

DATA CAPTURE NONE (None/Changes)

VALIDPROC NULL (NULL/Program name)

RESTRICT ON DROP . . . NO (Yes/No)

VOLATILE NO (Yes/No)

APPEND NO (Yes/No)

ALTER TABLE with any of the above changes OR select one of the options below

DROP FOREIGN KEY

ADD CHECK constraint

DROP CHECK constraint

ADD UNIQUE constraint

DROP UNIQUE constraint

ADD PERIOD

ACTIVATE ROW ACCESS CONTROL

DEACTIVATE ROW ACCESS CONTROL

ADD PARTITION

ADD CLONE

DROP CLONE

ADD VERSIONING

DROP VERSIONING

s ACTIVATE COLUMN ACCESS CONTROL

DEACTIVATE COLUMN ACCESS CONTROL

More: -

Figure 13-22 Activate column access control

Alternatively, you can use the ACT command against the column mask in the panel shown in Figure 13-25 on page 381, which generates the command shown in Example 13-6.

Example 13-6 Generated Column Access statement

```
ALTER TABLE "ADMR2"."CUSTOMER_COVERAGE" ACTIVATE COLUMN ACCESS CONTROL
```

Alternatively, you can use the ACT command from the Column Mask list, which generates the same statement for activating the mask (Figure 13-23).

```
ADB21PM n ----- VA1A Column Masks ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Line commands:
T - Table DO - Dep. objects DT - Dep. tables DC - Dep. cols C - Col
DF - Dep. functions ACT - Activate DACT - Deactivate RO - Role ENV - Envir
GEN - Generate DDL DDL - Object DDL CRE - CREATE DROP - Drop AL - Alter
? - Show all line commands

Sel Schema Name Table Schema Table name Column E E
* * * * * F I N S
* * * * *

ACT ADMR3 INCOME_BRANCH ADMR3 CUSTOMER INCOME A N N
***** END OF DB2 DATA *****
```

Figure 13-23 Activating a mask from the column list

Viewing column masks

Column masks can be viewed from two main paths. The first method is using the CM on the Table List panel. If there are any masks defined upon the table, the Column Mask panel opens. The second method is to use the CM option from the System Catalog panel (Figure 13-24). There are other panels from which you can access column masks, such as from a table column list if you know that the column is used in a mask, but you access the same panels regardless of where you enter the CM command.

```
ADB21 min ----- VA1A System Catalog ----- 15:27
Option ==>

More: +
DB2 System: VA1A
DB2 SQL ID: SYSADM

Authorization options:
OO - Object options
GA - Storage group auths
DA - Database authorizations
SA - Table space authorizations
TA - Table authorizations
VA - View authorizations
CA - Column authorizations
ZA - System authorizations
UA - User authorizations
RA - Resource authorizations
RO - Roles
CM - Column masks

PA - Plan authorizations
LA - Collection authorizations
KA - Package authorizations
HA - Schema authorizations
EA - User defined data type authorizations
FA - Function authorizations
OA - Stored procedure authorizations
QA - Sequence authorizations
TR - Trusted contexts
PM - Permissions

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name ==> INCOM% > Grantor ==> >
Owner ==> > Grantee ==> >
In D/L/H ==> > Switch Catalog Copy ==> N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
```

Figure 13-24 Object authorization options

After you enter CM and, optionally, using the selection criteria, the Column Mask panel (ADB21PM) opens (Figure 13-25).

```
ADB21PM n ----- VA1A Column Masks ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Line commands:
T - Table DO - Dep. objects DT - Dep. tables DC - Dep. cols C - Col
DF - Dep. functions ACT - Activate DACT - Deactivate RO - Role ENV - Envir
GEN - Generate DDL DDL - Object DDL CRE - CREATE DROP - Drop AL - Alter
? - Show all line commands

Sel Schema Name Table Schema Table name Column E E
* * * * * F I N S
* * * * *

-----
ADM3 INCOME_BRANCH ADM3 CUSTOMER INCOME A N N
***** END OF DB2 DATA *****
```

Figure 13-25 Column Mask panel

This panel lists all the column masks that meet the criteria entered in the selection panel. In our case, there is one mask, INCOME_BRANCH, which is specified on the ADMR3.CUSTOMER table. In this panel, you can navigate the DB2 catalog and view all the relevant information. If you want to view the CREATE statement for the mask, or model another mask upon this one, enter DDL, which generates the DDL and presents it in an ISPF edit session (Figure 13-26).

```
ISREDDE2 SYS10299.T153336.RA000.SYSADM.R0100129 Columns 00001 00072
Command ==> Scroll ==> CSR
***** Top of Data*****
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
000001 SET CURRENT SQLID='SYSADM';
000002 SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","SYSADM" ;
000003 CREATE MASK ADMR3.INCOME_BRANCH ON ADMR3.CUSTOMER
000004 FOR COLUMN INCOME RETURN
000005 CASE
000006 WHEN
000007 (VERIFY_GROUP_FOR_USER(SESSION_USER, 'ADM3', 'ADM1', 'ADM2') =
000008 1 AND (SUBSTR(CURRENT SQLID, 6, 1) = BRANCH)) THEN
000009 ADMR3.CUSTOMER.INCOME
000010 ELSE
000011 NULL
000012 END
000013 ENABLE ;
000014 COMMIT;
***** Bottom of Data*****
```

Figure 13-26 Mask definition

From here you can edit the definition and, optionally, run it when you exit the panel. Additional options on the panel (ADB21PM) allow you to view other information about the mask. For example, the DO option displays all dependent objects for the mask (Figure 13-27). This options shows that the column mask is dependant upon two columns (Income and Branch) in the ADMR3.CUSTOMER table.

ADB2DF in -- Browse Result of SQL Select							----- Row 1 to 2 of 2
Command ==>							Scroll ==> PAGE
L	BNAME	BSHEMA	BCOLNAME	BCOLNO	BTYPE	BOWNER	BOWNERTYPE DNAME
*	*	*		* *	*	*	*

	CUSTOMER	ADMR3	INCOME	4	C		INCOME_BRANCH
	CUSTOMER	ADMR3	BRANCH	5	C		INCOME_BRANCH
***** END OF DB2 DATA *****							

Figure 13-27 Column mask dependent objects

Figure 13-27 shows that the mask is dependent upon two objects, that is, both columns of table ADMR3.CUSTOMER, and that the mask name is INCOME_BRANCH. Another line command that is useful is the Interpret command, which produces a formatted output of the details of the mask (Figure 13-28).

ADB21PMI ----- VA1A Interpretation of an Object in SYSCONTROLS ----- 16:04	
Option ==>	
Details for mask . . : ADMR3.INCOME_BRANCH	
Schema : ADMR3	Name : INCOME_BRANCH
Owner : SYSADM	Owner type . . . : Auth ID
Table schema . . : ADMR3	Table name . . . : CUSTOMER
Column name . . : INCOME	Column number . : 4
Access control ID: 4	Created in DB2Ver: 0 - DB2 V10
Enforce : A - Always	Implicit : N - Explicitly created
Enabled : Y - Enabled	Status : ' ' - Complete
Environment ID . : 6	Created TS : 2010-10-26-13.33.30.524901
	Altered TS : 0001-01-01-00.00.00.000000
Search condition or expression (first 256 bytes):	
CASE WHEN (VERIFY_GROUP_FOR_USER(SESSION_USER, 'ADMR3', 'ADMR1', 'ADMR2') = 1 AND (SUBSTR(CURRENT SQLID, 6, 1) = BRANCH)) THEN	
ADMR3.CUSTOMER.INCOME	ELSE NULL END
Associated remarks :	

Figure 13-28 Interpret column masks

From the Tables, Views, and Aliases panel (ADB21T), you can immediately see if a table has any column mask, row permission, or both defined on it (Figure 13-29).

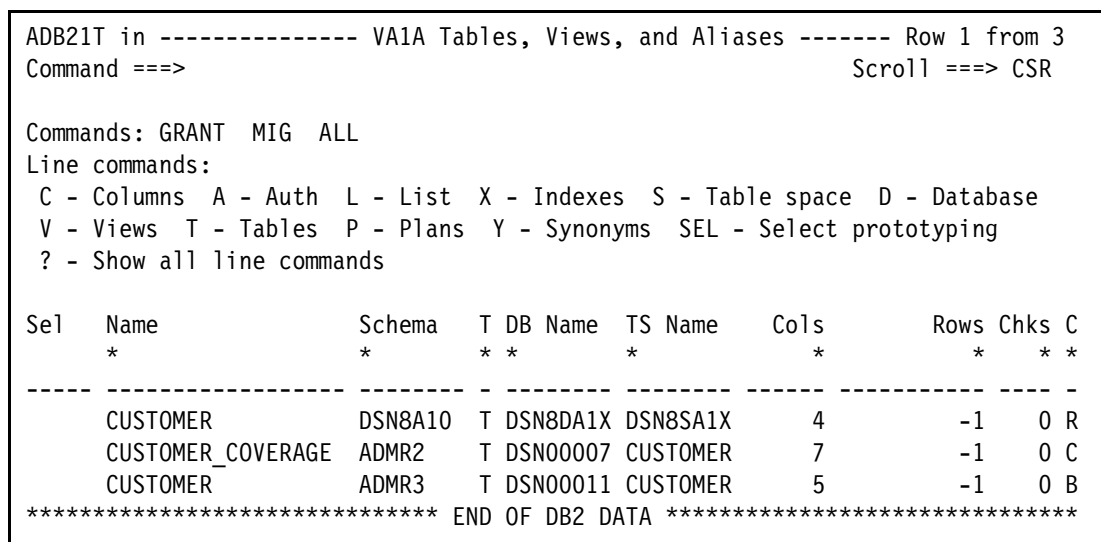


Figure 13-29 Showing permissions on a table list

Figure 13-29 shows a panel with three tables that have various permissions defined on them. The last column, C (access controls) shows that the first table has row permissions defined, the second table has a column mask defined, and the third table has both row permissions and column masks defined.

Deactivating column masks

There are two methods of deactivating a column mask:

- ▶ Through the Alter Table panel, shown in Figure 13-22 on page 379, except you select DEACTIVATE COLUMN ACCESS CONTROL instead.
- ▶ Through the DACT option from the Column Mask panel, shown in Figure 13-25 on page 381.

Both methods generate the SQL statement shown in Example 13-7.

Example 13-7 Deactivate a column mask

```
ALTER TABLE ADMR3.CUSTOMER DEACTIVATE COLUMN ACCESS CONTROL
```

This action removes the column access from the table but not the definition from the DB2 catalog, allowing you to use the ACTIVATE command to reinstate the command without recreating it.

Dropping column masks

When a column mask is no longer required, you can drop it. From the Column Mask List panel, issue the DROP line command. The Drop Column Mask panel opens (Figure 13-30).

```
ADB26DT n ----- VA1A Drop Column Mask ----- 11:15
Command ==>

DROP MASK

Schema . . . ADMR3      >
Name   . . . . INCOME_BRANCH      >
                                   (? to look up existing NONE)
```

Figure 13-30 Dropping column mask

This action generates the statement in Example 13-8.

Example 13-8 Drop column mask statement

```
DROP MASK "ADMR3"."INCOME_BRANCH"
```

13.5.2 Row permissions

A row permission is a database object that expresses a row access control rule for a specific table. It contains the row access rule as an SQL search condition that describes the condition under which the rows of a table can be accessed. If the row permission is not met, for example, the current SQLID does not match one defined in the permission, then no rows are displayed in the result set.

Creating a row permission

To create a row permission, you can, as in the case of Column Masks, use the DDL panel or you can use the PM option from the Authorization Object panel. From the DDL panel, you can select the option CPM. The Create Row Permission panel opens (Figure 13-31). This is the same panel that is reached by using the CRE option from the Row Permission List panel.

```
ADBP6CPM ----- VA1A Create Row Permission ----- 11:27
Command ==>

Commands: EDIT  COPY  CREATE

CREATE PERMISSION
Schema . . . . . > (default is SYSADM)
Name . . . . . > (? to look up)
ON (Table)
Schema . . . . . > (default is SYSADM)
Name . . . . . > (? to look up)
AS (Correlation)
Name . . . . . >
FOR ROWS WHERE
Search condition: (first 5 lines, use EDIT for all lines)
WHERE

ENFORCED FOR ALL ACCESS
ENABLE/DISABLE
Initial state . . DISABLE (Enable/Disable)
```

Figure 13-31 Initial Create Row Permission panel

Here we generate the row permissions that are referenced in *DB2 10 for z/OS Technical Overview*, SG24-7892. We enter the details defined in that book, except for the SEARCH (Figure 13-32).

```
ADBP6CPM ----- VA1A Create Row Permission ----- 11:37
Command ==>

Commands: EDIT  COPY  CREATE

CREATE PERMISSION
Schema . . . . . ADMR3   >      (default is SYSADM)
Name   . . . . . RA01_CUSTOMERS > (? to look up)
ON (Table)
Schema . . . . . ADMR3   >      (default is SYSADM)
Name   . . . . . CUSTOMER   > (? to look up)
AS (Correlation)
Name   . . . . .           >
FOR ROWS WHERE
Search condition:          (first 5 lines, use EDIT for all lines)
WHERE

ENFORCED FOR ALL ACCESS
ENABLE/DISABLE
Initial state . . DISABLE          (Enable/Disable)
```

Figure 13-32 Create Row Permission panel

This action enters the basic details of the permissions; to enter the search conditions, we use the EDIT command to open an ISPF Edit session where we will enter the search conditions (Example 13-9).

Example 13-9 Entering row permission search conditions

```
***** ***** Top of Data *****
000001 WHERE      ((CURRENT SQLID = 'DBOB#A'
000002           AND BRANCH = 'A')
000003           OR (CURRENT SQLID = 'DBOB#B'
000004           AND BRANCH = 'B')
000005           OR (CURRENT SQLID = 'DBOB#C'
000006           AND BRANCH = 'C')
000007           OR (CURRENT SQLID = 'DBOB##'))
***** ***** Bottom of Data *****
```

On returning to the previous panel, the search conditions are displayed (or partially displayed dependant upon the number of lines) (Figure 13-33).

```
ADBP6CPM ----- VA1A Create Row Permission ----- 11:43
Command ==>

Commands: EDIT  COPY  CREATE

CREATE PERMISSION
Schema . . . . . ADMR3    >      (default is SYSADM)
Name   . . . . . RA01_CUSTOMERS >  (? to look up)
ON (Table)
Schema . . . . . ADMR3    >      (default is SYSADM)
Name   . . . . . CUSTOMER   >  (? to look up)
AS (Correlation)
Name   . . . . .           >
FOR ROWS WHERE
Search condition:          (first 5 lines, use EDIT for all lines)
  WHERE ((CURRENT SQLID = 'DBOB#A'
    AND BRANCH = 'A')
    OR (CURRENT SQLID = 'DBOB#B'
    AND BRANCH = 'B')
    OR (CURRENT SQLID = 'DBOB#C'
  ENFORCED FOR ALL ACCESS
ENABLE/DISABLE
Initial state . . DISABLE          (Enable/Disable)
```

Figure 13-33 Create Row Permission panel with search condition

We can now create the row permission by selecting CREATE, which either runs the command directly or takes you to the Statement Execution Prompt panel (Figure 13-34).

```
ADB2PSTM ----- VA1A Statement Execution Prompt ----- 11:47
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'SYSADM.WSL.JCL'
Work statement list name ==> Action ==> A (Append or Replace)
More: +

Statement that is about to be executed (first 28 lines):
CREATE PERMISSION
"ADMR3"."RA01_CUSTOMERS"
ON "ADMR3"."CUSTOMER"

FOR ROWS
WHERE ((CURRENT SQLID = 'DB0B#A'
      AND BRANCH = 'A')
      OR (CURRENT SQLID = 'DB0B#B'
      AND BRANCH = 'B'))
```

Figure 13-34 Executing the row permission

The command that is executed is shown in Example 13-10.

Example 13-10 Generated row permission

```
CREATE PERMISSION
"ADMR3"."RA01_CUSTOMERS"
ON "ADMR3"."CUSTOMERS"
FOR ROWS WHERE
((CURRENT SQLID = 'DB0B#A'
  AND BRANCH = 'A')
  OR (CURRENT SQLID = 'DB0B#B'
  AND BRANCH = 'B')
  OR (CURRENT SQLID = 'DB0B#C'
  AND BRANCH = 'C')
  OR (CURRENT SQLID = 'DB0B##'))
ENFORCED FOR ALL ACCESS
DISABLE
```

As you can see, the permission is disabled. To activate the permission, refer to “Activating row permissions” on page 389.

If you are creating a row permission, you need to be aware if the table has any triggers defined on it and you need to change the trigger definition to ensure that they are defined as SECURE. This topic is discussed further in “Creating a column mask” on page 373 and describes how to use DB2 Administration Tool to alter the triggers.

Activating row permissions

A row permission is activated using the same panel that is used to activate a column mask (Figure 13-35). You can open this panel by issuing an AL(ter) command against the table. It is important to remember that as soon as the row permission is activated, DB2 imposes a default row permission predicate of 1=0 to all references of data to the table. This effectively restricts all access to the data in the table, as the condition can never be true. When there are multiple permissions, then DB2 joins the permissions using an “OR” condition, and data access is granted to the users who match the necessary conditions.

```

ADB21TA n ----- VA1A Alter Table ----- 11:50
Command ==>

Table schema . . : ADMR3      >
Table name . . . : CUSTOMER   >

AUDIT . . . . . NONE          (None, Changes, or All)
DATA CAPTURE . . . . . NONE    (None/Changes)
VALIDPROC . . . . . NULL       (NULL/Program name)
RESTRICT ON DROP . . . NO      (Yes/No)
VOLATILE . . . . . NO          (Yes/No)
APPEND . . . . . NO            (Yes/No)

ALTER TABLE with any of the above changes OR select one of the options below

DROP FOREIGN KEY          ADD PARTITION
ADD CHECK constraint      ADD CLONE
DROP CHECK constraint     DROP CLONE
ADD UNIQUE constraint     ADD VERSIONING
DROP UNIQUE constraint    DROP VERSIONING
ADD PERIOD                ACTIVATE COLUMN ACCESS CONTROL
s ACTIVATE ROW ACCESS CONTROL  DEACTIVATE COLUMN ACCESS CONTROL
DEACTIVATE ROW ACCESS CONTROL
  
```

Figure 13-35 Activate row access control

This action generates the command shown in Example 13-11.

Example 13-11 Generated row access statement

```
ALTER TABLE "ADMR3"."CUSTOMER" ACTIVATE ROW ACCESS CONTROL
```

Alternatively, you can use the ACT command on the Row Permissions panel to activate this control (Figure 13-36).

```
ADB21PM n ----- VA1A Row Permissions ----- Row 1 to 1 of 1
Command ==> Scroll ==> CSR

Line commands:
T - Table  DO - Dep. objects  DT - Dep. tables  DC - Dep. cols
DF - Dep. functions  ACT - Activate  DACT - Deactivate  RO - Role  ENV - Envir
GEN - Generate DDL  DDL - Object DDL  CRE - CREATE  DROP - Drop  AL - Alter
? - Show all line commands

Sel  Schema      Name              Table              O E  E
      *          *              Schema      Table name      *  T F I N S
-----
ACT  ADMR3      RA01_CUSTOMERS  ADMR3      CUSTOMER          SYSADM      A N N
***** END OF DB2 DATA *****
```

Figure 13-36 Activating row permissions from Row Permissions panel

Viewing row permissions

Like column masks, row permissions can be viewed using either of two methods. The first method for viewing the permissions is to use the PM command from the table list (Figure 13-37).

```
ADB21T in ----- VA1A Tables, Views, and Aliases ---- Row 1 to 4 of 4
Command ==> Scroll ==> CSR

Commands: GRANT  MIG  ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables  P - Plans  Y - Synonyms  SEL - Select prototyping
? - Show all line commands

Sel  Name              Schema  T DB Name  TS Name  Cols      Rows Chks C
      *              *      *  *      *      *      *      *  *
-----
      CUSTOMER          DSN8A10 T DSN8DA1X DSN8SA1X   4        -1    0
      CUSTOMERS         DSN8A10 T DSN8DA1Y DSN8SA1Y  12         5    0
      CUSTOMER_COVERAGE ADMR2   T DSN00007 CUSTOMER   7        -1    0
PM    CUSTOMER          ADMR3   T DSN00011 CUSTOMER   5        -1    0 R
***** END OF DB2 DATA *****
```

Figure 13-37 Row permissions from a table list

If there are any permissions (we know there is at least one permission defined due to the presence of the R on the table detail line), then the Permissions panel opens (Figure 13-38).

```
ADB21PM n ----- VA1A Permissions ----- Row 1 to 3 of 3
Command ==>                                     Scroll ==> PAGE

Line commands:
T - Table  DO - Dep. objects  DT - Dep. tables  DC - Dep. cols
DF - Dep. functions  ACT - Activate  DACT - Deactivate  RO - Role  ENV - Envir
GEN - Generate DDL  DDL - Object DDL  CRE - CREATE  DROP - Drop  AL - Alter
? - Show all line commands

Sel  Schema      Name      Table
      *          *          *      Schema      Table name      Owner      O E  E
      *          *          *      *          *          *      T F I N S
      *          *          *      *          *          *      * * * * *
----->-----
      ADMR3      RA01_CUSTOMERS  ADMR3      CUSTOMER      SYSADM      A N Y
      ADMR3      RA02_CUSTOMERS  ADMR3      CUSTOMER      SYSADM      A N Y
      ADMR3      SYS_DEFAULT_ROW_PE  ADMR3      CUSTOMER      SYSIBM      A Y Y
***** END OF DB2 DATA *****
```

Figure 13-38 Row permissions list

Note that the third line is the system default row; this permission is automatically inserted by DB2. As per the column masks, you have a variety of commands for traversing the catalog, for finding dependent tables and columns, and so on. You can also generate the definition for the permission, as you can for column masks, as shown in Figure 13-39. This command produces the DDL for the permission with any requested changes through the input fields. You could also use the DDL command as per the example for column masks.

```

ADB2GENS ----- VA1A Generate SQL from DB2 catalog -----14:07
Option ==>

Generate SQL statements for:                                DB2 System: VA1A
permission ADMR3.RA02_CUSTOMERS                            DB2 SQL ID: SYSADM
                                                            More:      +

SQL statement types to be generated from the DB2 catalog:

PERMISSION . . . . . Y (Y,N)                                COMMENT ON . . . . . Y (Y,N)

New names/values for generated SQL: (leave blank to use current values)
Object owner . . . . . > Run SQLID . . . . .
Object grantor . . . . . >
Object schema . . . . . >
Target DB2 version . . . . . (Current DB2 version: 1015)
Use Masking . . . . . NO (Yes/No)
Generate catalog stats . NO (Yes/No/Only)
Target cat qualifier . > (Default is SYSIBM)

Output file and execution mode:
Add to work stmt list . NO (Yes/No)
Data set name . . . . .
Data set disposition . OLD (OLD, SHR, or MOD)

```

Figure 13-39 Generate a row permission

The interpret output for the row permissions is similar to the interpret output for the column masks shown in Figure 13-28 on page 382, except that the column name and number have been removed, as they are not relevant for row permissions. You can see the environmental variables in use at the time the permission was created by using the ENV command. This command opens the Environment Variables panel (Figure 13-40).

```

ADB21ENV ----- VA1A Environment Variables ----- Row 1 to 1 of 1
Command ==>                                           Scroll ==> PAGE

Line commands:
I - Interpretation

Application      Original Host      D M S S M D D T F C
Sel Schema      CCSID      CCSID Language P S D S D A F F S F R
*              *              * *      * * * * * * * * * *

-----
SYSADM          37          37 NONE      P N A A N 1 J I S A N E
***** END OF DB2 DATA*****

```

Figure 13-40 Row permission Environmental Variables panel

There is a help panel for this panel that explains the variables and values; alternatively, use the Interpret command to see a more descriptive panel of the variables (Figure 13-41).

```
ADB21EVI ----- VA1A Interpretation of an object in SYSENVIRONMENT -----14:18
Option ==>

Details for Permission: ADMR3.RA02_CUSTOMERS

Environment ID . . . : 6                      Application CCSID. . . : 37
Current Schema . . . : SYSADM                 Original CCSID . . . : 37
Created in DB2 Ver : 0 - DB2 V10             String delimiter . . . : Apostrophe
Decimal point. . . : P - Period              SQL String delimiter : Apostrophe
Min Divide scale . . : No                    Uses Mixed data. . . : No
Float format . . . : System/390              Decimal Arithmetic . . : 1
Fold . . . . . : No                         Date format. . . . . : JIS
Host language. . . : NONE                   Time format. . . . . : ISO
Rounding mode. . . : Round Half Even        Char set . . . . . : Alphanumeric
Schema Path . . . : "SYSIBM","SYSFUN","SYSPROC","SYSADM"
```

Figure 13-41 Interpreting the ENV variables

There is a similar function for column masks.

Deactivating row permissions

To deactivate a row permission, you can either issue the AL(ter) table command and use the panel shown in Figure 13-22 on page 379 (select DEACTIVATE ROW ACCESS CONTROL), or use the DACT command on the Permissions panel (Figure 13-38 on page 391). Both methods generate the command shown in Example 13-12.

Example 13-12 Deactivate row permissions

```
ALTER TABLE ADMR3.CUSTOMER DEACTIVATE ROW ACCESS CONTROL
```

When you deactivate row access controls, DB2 removes the system default permission (1=0), which allows all users to access the rows again, as shown by the row permissions in the catalog after deactivation (Figure 13-42). Note that the user defined row permissions are still available for reactivation.

```
ADB21PM n ----- VA1A Permissions ----- Row 1 to 2 of 2
Command ==>                                     Scroll ==> PAGE

Line commands:
T - Table DO - Dep. objects DT - Dep. tables DC - Dep. cols
DF - Dep. functions ACT - Activate DACT - Deactivate RO - Role ENV - Envir
GEN - Generate DDL DDL - Object DDL CRE - CREATE DROP - Drop AL - Alter
? - Show all line commands

Sel Schema      Name      Table      Table name      Owner      O E  E
      *          *          *          *              *      T F I N S
-----
      ADMR3      RA01_CUSTOMERS  ADMR3      CUSTOMER        SYSADM      A N Y
      ADMR3      RA02_CUSTOMERS  ADMR3      CUSTOMER        SYSADM      A N Y
*****
                                END OF DB2 DATA *****
```

Figure 13-42 Row permissions after deactivation

Dropping row permissions

When a row permission is no longer required, you can drop it as you would any other object. From the Row Permission List panel, issue the DROP line command. The Drop Row Permission panel opens (Figure 13-43).

```
ADB26DT n ----- VA1A Drop Row Permission ----- 12:14
Command ==>

DROP PERMISSION

Schema . . . ADMR3      >
Name   . . . . RA01_CUSTOMERS  >
                                (? to look up existing NONE)
```

Figure 13-43 Drop Row Permission panel

This action creates the statement shown in Example 13-13.

Example 13-13 Generated drop row permission

```
DROP PERMISSION "ADMR3"."RA01_CUSTOMERS"
```

13.5.3 Trusted context

A trusted context addresses the problem of establishing a trusted relationship between DB2 and an external entity, for example, an application server. A trusted context is a series of trusted attributes that are evaluated to determine whether a specific context can be trusted. When the trusted context is established, users can be enabled switch to another user ID, but only within the established trusted context.

In addition, you can assign a role (see 13.5.4, “Roles” on page 406) to a user of a trusted context, allowing that user to assume extra authorities while connected through the trusted context.

Creating a trusted context

For our example, we are creating two trusted contexts. The first allows a remote server to connect for a specific user. The second is for a local trusted connection for a batch job.

In 13.5.4, “Roles” on page 406, we add ROLES to the trusted contexts to allow specific actions to be undertaken from the trusted context.

First, define a trusted context (ADMR3USR) from a remote server (IP address 3.11.222.255) using ADMR3 authorization ID. From the main authorization panel, select option TR - trusted context. The Trusted Contexts panel opens (Figure 13-44).

ADB2AN in ----- DB0B Trusted Contexts ----- Row 1 to 4 of 4
Command ==> Scroll ==> PAGE

Line commands:
RO - Roles ID - Authids ATTR - Attributes DR - Definer role
I - Interpretation DROP - Drop COM - Comment CRE - Create AL - Alter
ADDA - Add attribute ADDI - Add id DDL - Generate DDL GEN - Generate SQL

Se1	Name	Definer	D System T Authid	Default Role	O E A A T N L U	Created Timestamp
*	*	*	*	*	*	*
----->	----->	----->	----->	----->	----->	----->
	CTXSYSAD	DB2R5	DB2R5	DB0BSYSA	Y N N	2010-08-08-12.25.27.112041
	CTXSYSDB	DB2R5	DB2R54	DB0BSYSD	L Y N N	2010-08-08-13.23.13.969587
	CTX_DJ	DB0BSECA	DB2R51		Y N N	2010-08-18-03.46.36.498101
	CTXIDID	DB2R5	DB2R53	DB0BEP	Y N N	2010-08-26-01.29.28.341121
***** END OF DB2 DATA *****						

Figure 13-44 Trusted Contexts panel

Enter the CRE line command and the details of the trusted context. The Create Trusted Context panel opens (Figure 13-45).

```
ADB26CN n ----- DB0B Create Trusted Context ----- 17:42
Command ==>

CREATE TRUSTED CONTEXT
Name . . . . . admr3usr > (? to look up existing)

BASED UPON CONNECTION USING SYSTEM AUTHID
Authid . . . . . admr3 > (Primary Authid)

DEFAULT ROLE
Role . . . . . > (role name)

WITH ROLE AS OBJECT OWNER AND QUALIFIER
With owner/qual . . . (Yes/No)

ENABLE/DISABLE
Initial state . . . ENABLE (Enable/Disable)

DEFAULT SECURITY LABEL
Label . . . . . (security label name)

(continued...)

Press ENTER to continue with attributes or PF3 to cancel.
```

Figure 13-45 Creating a trusted context: Part 1

Press Enter. Another Create Trusted Context panel opens (Figure 13-46).

```
ADB26CNA ----- DB0B Create Trusted Context ----- 17:58
Command ==>

CREATE TRUSTED CONTEXT "CTXIDID"

ATTRIBUTES (

Choose one:

ADDRESS . . . 3.11.222.255      >                (IP address)

ENCRYPTION . .                  (None, Low, or High)

SERVERAUTH . .                  >                (network security zone)
                                           (jobname or job prefix*)

JOBNAME . . .

    Add more attributes
    )

Press ENTER to continue with IDs or PF3 to restart attribute definition.
```

Figure 13-46 Creating a trusted context: Part 2

For this example, enter the IP address, 3.11.222.255. The Create Trusted Context IDs panel opens (Figure 13-47).

```
ADB26CNU ----- DB0B Create Trusted Context IDs ----- 18:03
Command ==>

More:      +

CREATE TRUSTED CONTEXT "CTXIDID"
ATTRIBUTES(. . .)

WITH USE FOR (

Choose one:

Authorization name . . . . . (primary authid for context)
EXTERNAL SECURITY PROFILE . . (profile name)
PUBLIC . . . . . (Yes, No - optional)

WITH AUTHENTICATION option . . (Yes, No - optional)

User options for Authorization name and Security profile:
ROLE . . . . . (role name)
SECURITY LABEL . . . . . (security label name)

Add more users
    )
```

Figure 13-47 Creating a trusted context: Part 3

On this panel, you can add any additional attributes for the trusted context. Notice that you can add the role at this stage. Alter the trusted context to add the role alter. Press Enter, and the command shown in Example 13-14 is generated and executed.

Example 13-14 Generated trusted context

```
CREATE TRUSTED CONTEXT "ADMR3USR"
  BASED UPON CONNECTION USING SYSTEM AUTHID ADMR3
  NO DEFAULT ROLE
  ENABLE
  NO DEFAULT SECURITY LABEL
  ATTRIBUTES( ADDRESS '3.11.222.255')
```

The first trusted context is now created. The second trusted context will be used to secure DBA activities. We have a DBA implementing changes to a customer data base through a trusted context and a role. For this example, we assume that the role has been previously created, (CUSTDBA), the user name is ADMR2, and it is the OWNER of the objects, as shown in the Create Trust Context panel (Figure 13-48). Press Enter.

```
ADB26CN n ----- DB0B Create Trusted Context ----- 19:53
Command ==>

CREATE TRUSTED CONTEXT
Name . . . . . DB0BDBA > (? to look up existing)

BASED UPON CONNECTION USING SYSTEM AUTHID
Authid . . . . . ADMR2 > (Primary Authid)

DEFAULT ROLE
Role . . . . . custdba > (role name)

WITH ROLE AS OBJECT OWNER AND QUALIFIER
With owner/qual . . YES      (Yes/No)

ENABLE/DISABLE
Initial state . . . DISABLE   (Enable/Disable)

DEFAULT SECURITY LABEL
Label . . . . .              (security label name)

(continued...)

Press ENTER to continue with attributes or PF3 to cancel.
```

Figure 13-48 Create a local trusted context: Part 1

Add the JOBNAME parameter (Figure 13-49) and press Enter.

```

ADB26CNA ----- DBOB Create Trusted Context ----- 20:11
Command ==>

CREATE TRUSTED CONTEXT "CTXIDID"

  ATTRIBUTES (

    Choose one:

    ADDRESS . . . > (IP address)

    ENCRYPTION . . (None, Low, or High)

    SERVERAUTH . . > (network security zone)
                     (jobname or job prefix*)

    / Add more attributes
      )

Press ENTER to continue with IDs or PF3 to restart attribute definition.

```

Figure 13-49 Create a local trusted context: Part 2

You can have multiple values for these attributes within one trusted context. To add more values, select Add more attributes (Figure 13-49). We create the ADMR2BCH jobname.

JOBNAME can have three different meanings, depending upon the type of connection (Table 13-2).

Table 13-2 Jobname attributes

Source of the address space	Jobname
RRSAF	Job name or started task name
TSO	TSO LOGON ID
BATCH	JOBNAME on JOB statement

The statement that is created from these panels is shown in Example 13-15.

Example 13-15 Generated local trusted context

```

CREATE TRUSTED CONTEXT "DBOBDDBA"
  BASED UPON CONNECTION USING SYSTEM
  AUTHID ADMR2
  DEFAULT ROLE CUSTDBA
  WITH ROLE AS OBJECT OWNER AND QUALIFIER
  DISABLE
  NO DEFAULT SECURITY LABEL
  ATTRIBUTES( JOBNAME 'ADMR2', JOBNAME 'ADMR2BCH')

```

This statement allows ADMR2 to assume the CUSTDBA role (which has the necessary authorities) when connecting through TSO, but all changes made through batch must have a jobname of ADMR2BCH. This trusted context is currently disabled and is not used; it can be turned on and off as required to ensure that the authorities are only present when you need to make a change.

Specific users often need to be given specific authorizations, which can be achieved by using trusted contexts. You can use the “WITH USE” syntax to provide a trusted context to a subset of users. To enter the users that you want to have those authorizations, enter the values shown in Figure 13-47 on page 397. Add the security criteria that you want to enter, in this case ADMR1, which uses a different role, and ADMR5, which uses the default role and needs authenticating.

First, add ADMR1 (Figure 13-50). This user is using the PRODDBA role. Notice that we have selected to add more users.

```

ADB26CNU ----- DB0B Create Trusted Context IDs -----14:27
Command ==>

                                                                    More:      +

CREATE TRUSTED CONTEXT "DB0BDBA"
ATTRIBUTES(. . .)

WITH USE FOR (

Choose one:

  Authorization name . . . . . admr1      (primary authid for context)
  EXTERNAL SECURITY PROFILE . .      (profile name)
  PUBLIC . . . . .      (Yes, No - optional)

WITH AUTHENTICATION option . .      (Yes, No - optional)

User options for Authorization name and Security profile:
  ROLE . . . . . PRODDBA      (role name)
  SECURITY LABEL . . . . .      (security label name)

s Add more users
  )

```

Figure 13-50 Adding users to a trusted context

The Create Trusted Context IDs panel opens again and we enter the second user's details (Figure 13-51).

```

ADB26CNU ----- DB0B Create Trusted Context IDs ----- 14:29
Command ==>

                                                                    More:  +

CREATE TRUSTED CONTEXT "DB0BDBA"
ATTRIBUTES(. . .)

WITH USE FOR (

Choose one:

    Authorization name . . . . . ADMR5      (primary authid for context)
    EXTERNAL SECURITY PROFILE . .      (profile name)
    PUBLIC . . . . .      (Yes, No - optional)

WITH AUTHENTICATION option . . yes      (Yes, No - optional)

User options for Authorization name and Security profile:
    ROLE . . . . .      (role name)
    SECURITY LABEL . . . . .      (security label name)

    Add more users
    )

```

Figure 13-51 Adding additional users to the trusted context

This actions creates the CREATE statement (Example 13-16).

Example 13-16 Generated local trusted context with users

```

CREATE TRUSTED CONTEXT "DB0BDBA"
  BASED UPON CONNECTION USING SYSTEM
  AUTHID ADMR2
  DEFAULT ROLE CUSTDBA
  WITH ROLE AS OBJECT OWNER AND QUALIFIER
  DISABLE
  NO DEFAULT SECURITY LABEL
  ATTRIBUTES( JOBNAME 'ADMR2', JOBNAME 'ADMR2BCH')
  WITH USE FOR  ADMR1 ROLE PRODDBA
, ADMR5  WITH AUTHENTICATION

```

User ADMR1 now uses the PRODDBA role when using the trusted context, while ADMR5 uses the default role of CUSTDBA and needs to be authenticated when connecting to DB2.

Altering a trusted context

If the trusted context already exists, then you can alter the context to change the contents or to add extra options. To amend any of the values shown in Figure 13-48 on page 398, use the AL command and that panel opens. You can amend those values and your changes will be made (no further panels will open).

If you want to add attributes, you could issue the ADDA command, which opens the Alter Trusted Context panel (Figure 13-52).

```
ADB26CNA ----- DB0B Alter Trusted Context ----- 17:57
Command ==>

ALTER TRUSTED CONTEXT "DB0BDBA"

ADD ATTRIBUTES (

Choose one:

ADDRESS . . . > (IP address)

SERVERAUTH . . > (network security zone)
                  (jobname or job prefix*)
JOBNAME . . .

    Add more attributes
    )

Press ENTER to continue with IDs or PF3 to restart attribute definition.
```

Figure 13-52 Adding Trusted Context Attribute

This panel is similar to the panels used in the creation process (see Figure 13-49 on page 399), except that the encryption selection is not available, as this is set at the creation time.

A better way is to use the ATTR line command, which opens you a panel where you can add, drop, or alter the attributes, including the encryption level for the trusted context (Figure 13-53).

```
ADB2ANAT --- DB0B Attributes for Trusted Context DB0BDBA      Row 1 to 2 of 2
Command ==>                                                    Scroll ==> PAGE

Line commands:
DROP - Drop  AL - Alter  ADD - Add  I - Interpretation

Sel  Name          Value          Created Timestamp
    *              *              *
-----
      JOBNAME      ADMR2          2010-10-28-20.15.10.067659
      JOBNAME      ADMR2BCH       2010-10-28-20.15.10.067659
***** END OF DB2 DATA *****
```

Figure 13-53 Altering or adding attributes to a trusted context

If you issue ADD, the Alter Trusted Context panel opens (Figure 13-52). If you issue an AL command, the same panel opens, but with the attribute features populated. Change the value and DB2 Administration Tool builds the statements to run the change.

If you issue the DROP command, the Drop Attributes panel opens and prompts you to confirm your choices (Figure 13-54).

```
ADB26DR n ----- DB0B Drop Attributes ----- 19:09
Command ==>

ALTER TRUSTED CONTEXT

Name . . . . DB0BDBA          >

DROP ATTRIBUTES (

ATTRIBUTE NAME . . . JOBNAME          >
ATTRIBUTE VALUE . . ADMR3              > (? to look up)

)
```

Figure 13-54 Dropping an attribute from a trusted context

This action generates the DROP statement (Example 13-17).

Example 13-17 Drop attribute statement

```
ALTER TRUSTED CONTEXT "DB0BDBA" DROP ATTRIBUTES ( JOBNAME 'ADMR3')
```

The final attribute that you may want to add or alter is the USE attribute. The USE attribute allows you to add specific users to the trusted context to alter their connection properties.

Again, there are two ways of adding a user:

- ▶ Use ADDI on the main Trusted Context panel
- ▶ Issue the ID line command, which opens the Authids for Trusted Context panel (Figure 13-55). This is the preferred option, as this panels gives you the ability add a user, drop a user, or replace the attributes of an existing user.

```
ADB2ANID ----- DB0B Authids for Trusted Context DB0BDBA - Row 1 to 1 of 1
Command ==>                                           Scroll ==> PAGE

Line commands:
RO - Role  DROP - Drop ID  ADD - Add ID  REPL - Replace ID  I - Interpretation

Sel  Authid  Role    A Created Timestamp      Security Label
    *        *      * *
-----
      ADMR3          N 2010-10-29-19.26.00.591528
***** END OF DB2 DATA *****
```

Figure 13-55 Trusted Context IDs

To add a user, issue the ADD command, which opens the Create Trusted Context IDs panel (Figure 13-50 on page 400). If you want to replace attributes of a user, issue the REPL command, which opens the Alter Trust Context IDs panel (Figure 13-56). In this example, you are turning on the authentication attribute.

ADB26CNU ----- DB0B Alter Trusted Context IDs -----		19:40
Command ==>		
		More: +
ALTER TRUSTED CONTEXT "DB0BDBA"		
REPLACE USE FOR (
Choose one:		
Authorization name	ADMR3	(primary authid for context)
EXTERNAL SECURITY PROFILE . .		(profile name)
PUBLIC		(Yes, No - optional)
WITH AUTHENTICATION option . .	yes	(Yes, No - optional)
User options for Authorization name and Security profile:		
ROLE		(role name)
SECURITY LABEL		(security label name)
)		

Figure 13-56 Altering attributes of an ID

This action generates the ALTER command shown in Example 13-18.

Example 13-18 Generated Replace command

```
ALTER TRUSTED CONTEXT DB0BDBA
REPLACE USE FOR  ADMR3  WITH AUTHENTICATION
```

If you issue the DROP command, the command shown in Example 13-19 is generated. Notice that if you do not have prompts switched on, then this DROP happens immediately; there are no drop impact reports.

Example 13-19 Drop ID command

```
ALTER TRUSTED CONTEXT "DB0BDBA" DROP USE FOR "ADMR3"
```

Viewing a trusted context

To view a trusted context, you need to use the TR option from the Authorization options panel. From this panel, you can either just enter TR and see all trusted contexts or you can narrow the results set by entering a name or partial name. The panel shown in Figure 13-57 lists all the trusted contexts that are defined in the subsystem.

```

ADB2AN in ----- DB0B Trusted Contexts ----- Row 1 to 6 of 6
Command ==> Scroll ==> PAGE

Line commands:
RO - Roles ID - Authids ATTR - Attributes DR - Definer role
I - Interpretation DROP - Drop COM - Comment CRE - Create AL - Alter
ADDA - Add attribute ADDI - Add id DDL - Generate DDL GEN - Generate SQL

Sel  Name      Definer  D System  Default  O E A A Created
      *         *       T Authid  Role     T N L U Timestamp
-----> -----> -----> -----> -----> ----->
      CTXSYSAD DB2R5      DB2R5     DBOBSYSA Y N N 2010-08-08-12.25.27.112041
      CTXSYSDB DB2R5      DB2R54    DBOBSYSD L Y N N 2010-08-08-13.23.13.969587
      CTX_DJ    DBOBSECA  DB2R51    Y N N 2010-08-18-03.46.36.498101
      CTXIDID   DB2R5      DB2R53    DBOBEP    Y N N 2010-08-26-01.29.28.341121
      ADMR3USR  ADMR3      ADMR3      Y N N 2010-10-28-18.16.41.901432
      DBOBDDBA  ADMR3      ADMR2      CUSTDBA   L Y N N 2010-10-28-20.15.10.067659
***** END OF DB2 DATA*****

```

Figure 13-57 Trusted context list

From this panel, you can either enter the GEN or DDL command. The DDL command shows you the DDL Forth object in an ISPF edit session, while GEN gives you the option of generating the DDL in either batch or online. As trusted contexts are independent database objects, there are no dependant objects.

Enabling and disabling a trusted context

To enable a trusted context, issue the AL(ter) command against the trusted context. The Create Trusted Context panel opens (Figure 13-48 on page 398). Change DISABLE to ENABLE and the ALTER statement (Example 13-20) will be generated and executed.

Example 13-20 Enable a trusted context

```
ALTER TRUSTED CONTEXT "DBOBDDBA"
  ALTER SYSTEM AUTHID ADMR2
  DEFAULT ROLE CUSTDBA
  WITH ROLE AS OBJECT OWNER AND QUALIFIER
  ENABLE
  NO DEFAULT SECURITY LABEL
```

To disable the trusted context, simply reverse the process.

13.5.4 Roles

As you can see from the trusted context examples, you can extend the power of the trusted context by using *roles*. A role, which is only available through a trusted context, is a database entity that groups together one or more privileges. Roles give you the ability to give users a means to acquire context specific authorizations. By implementing roles, you can give ownership of an object to multiple authids, rather than one authid owning the objects. A role can also be the subject of auditing.

Creating a role

To create a role, you need to navigate to the Create/Drop/Label/Comment On Objects panel (ADB26) (Figure 13-58). To open this panel, either select option 2.4 from the DB2 Administration Menu panel or enter DDL from a menu panel.

```
ADB26 min ----- VA1A Create/Drop/Label/Comment On Objects ----- 12:18
Option ==>

More:      +
DB2 System: VA1A
DB2 SQL ID: SYSADM

CREATE      DROP
CG - Storage group      DG - Storage group
CD - Database           DD - Database
CS - Table space        DS - Table space
CT - Table              DT - Table
CV - View               DV - View
CL - Alias              DL - Alias
CX - Index              DX - Index
CY - Synonym            DY - Synonym
CA - Auxiliary table
CE - Distinct type      DE - Distinct type
CJ - Trigger            DJ - Trigger
CF - Function           DF - Function
CO - Stored procedure    DO - Stored procedure
CM - Materialized table
CQ - Sequence           DQ - Sequence
CTR - Trusted context    DTR - Trusted context
CRO - Role              DRO - Role
CCM - Column mask        DCM - Column mask
CPM - Row permission     DPM - Row permission
LABEL        COMMENT (remark)
LT - Table/view          RT - Table/view
LL - Alias               RL - Alias
LC - Column              RC - Column
                        RX - Index
                        RQ - Sequence
                        RTR - Trusted context
                        RRO - Role
                        RCM - Column mask
                        RPM - Row permission
```

Figure 13-58 Create/Drop/Label/Comment On Objects panel

From this panel, you can issue the CRO command, which opens the Create Role panel (Figure 13-59).

```
ADB26CRO ----- DB0B Create Role-----
13:44
Command ==>

CREATE ROLE

Name . . . . CUSTDBA          > (? to look up existing)
```

Figure 13-59 Creating a role

Press Enter and the role is created. You may need to grant authorization to the role.

Viewing a role

A role can be viewed from many of the DB2 Administration Tool panels. You can issue the RO command from any panel where the object could have been created by a role. To discover if an object is created by a role, you need to interrogate the catalog to see if either CREATORTYPE = 'L', OWNERTYPE = 'L', GRANTORTYPE = 'L', or GRANTEETYPE = 'L' exist. When you enter the RO or RR command against an object or authority associated with a role, the Roles panel (ADB2ARL) opens (Figure 13-60).

```
ADB2ARL n ----- DB0B Roles ----- Row 1 to 1 of 1
Command ==>                                     Scroll ==> PAGE

Line commands:
TC - Trusted Contexts  OBJ - Dependent objects  DROP - Drop  COM - Comment
DR - Definer role     I - Interpretation  DDL - Generate DDL  GEN - Generate SQL

Sel  Name      Definer  T Created Timestamp
    *          *        * *
-----
*    CUSTDBA   ADMR3     2010-10-28-19.52.14.472606
***** END OF DB2 DATA *****
```

Figure 13-60 Viewing roles

From this panel, you can navigate to the trusted context(s) associated with this role by using the TC line command (Figure 13-61).

```
ADB2AN in ----- DB0B Trusted Contexts ----- Row 1 to 2 of 2
Command ==>                                     Scroll ==> PAGE

Line commands:
RO - Roles  ID - Authids  ATTR - Attributes  DR - Definer role
I - Interpretation  DROP - Drop  COM - Comment  CRE - Create  AL - Alter
ADDA - Add attribute  ADDI - Add id  DDL - Generate DDL  GEN - Generate SQL

Sel  Name      Definer  D System  Default  O E A A Created
      *         *      *  *      *      * * * * *
-----
      ADMR3USR ADMR3      ADMR3  CUSTDBA  N N N 2010-10-28-18.16.41.901432
      DBOBDBA ADMR3      ADMR2  CUSTDBA  L Y N 2010-10-28-20.15.10.067659
*****
                                END OF DB2 DATA *****
```

Figure 13-61 From role to trusted context

From this panel, you can issue the GEN primary command to reverse engineer all the trusted contexts that use this role (Example 13-21).

Example 13-21 Generate trusted context used by a role

```
-----
--
-- Database 2 Administration Tool (DB2 Admin) , program 5655-W34 --
--
-- ADB2GEN - Extract object definitions from the DB2 Catalog tables --
--
-- Input prepared on : DB0B (1015 )   Extract time : 2010-10-30 15:16 --
--
-- Catalog values overridden : none
--
-- Generate : SG=N DB=N TS=N TB=N VW=N IX=N SY=N AL=N LB=Y CM=Y FK=N --
--           TG=N UT=Y UF=Y SP=Y SQ=Y RO=N TC=Y MK=Y PM=Y AC=Y --
-- Grants   : SG=N DB=N TS=N TB=N VW=N SC=N UT=N UF=Y SP=Y SQ=N --
--
--
-- SET CURRENT SQLID='ADMR3';
--
-- ADB2GEN: Generate DDL for Trusted Context ADMR3USR
--
--
--
-- Trusted Context=ADMR3USR
--
--
-- CREATE TRUSTED CONTEXT ADMR3USR
--   BASED UPON CONNECTION USING SYSTEM AUTHID ADMR3
--   DEFAULT ROLE CUSTDBA
```

```

        WITHOUT ROLE AS OBJECT OWNER
        DISABLE
        NO DEFAULT SECURITY LABEL
        ATTRIBUTES (
        ADDRESS '3.11.222.255',
        ENCRYPTION 'NONE'
        ) ;
--
    COMMIT;
-----
--
-- ADB2GEN: Generate DDL for Trusted Context DBOBDBA
--
-----
--
--
-----
-- Trusted Context=DBOBDBA
-----
--
--
CREATE TRUSTED CONTEXT DBOBDBA
    BASED UPON CONNECTION USING SYSTEM AUTHID ADMR2
    DEFAULT ROLE CUSTDBA
    WITH ROLE AS OBJECT OWNER AND QUALIFIER
    ENABLE
    NO DEFAULT SECURITY LABEL
    ATTRIBUTES (
    JOBNAME 'ADMR2BCH',
    JOBNAME 'ADMR2',
    JOBNAME 'ADMR3'
    )
    WITH USE FOR
    ADMR3
    WITHOUT AUTHENTICATION ;
--
    COMMIT;
--
-----
-- ADB2GEN - End of generated DDL
-----

```

Dropping a role

To drop a role, issue the DROP command against the role from within the Role panel. This action opens the Drop Role panel (Figure 13-62). The role is dropped when you press Enter.

```

ADB26DR n ----- DBOB Drop Role ----- 16:00
Command ==>

DROP ROLE

Name . . . . CUSTDBA          > (? to look up)

```

Figure 13-62 Dropping a role

13.6 Auditing profiles

DB2 10 for z/OS assists you with monitoring access to your data by providing a more powerful and flexible audit capability that is based upon audit policies. These policies, when used with the audit filtering options, provide an auditing solution that helps you monitor application and user data access, including access from administrative authorities.

DB2 Administration Tool can help you define these policies and administrate changes to the policies.

All auditing profile are accessed from the DB2 Systems Administration panel. Using the AP option opens the Manage Audit Policies panel (Figure 13-63), which lists all the audit policies defined within the DB2 subsystem.

```
ADBPZAP n ----- DB0B Manage Audit Policies ----- Row 1 to 14 of 14
Command ==> Scroll ==> PAGE

Line commands:
I - Interpret    U - Update    INS - Insert    D - Delete    S - Show object

Sel  Name      Object      Object      C V O E C S      D
   *      *      *      T H A M X O M SYSAD DBADM Database Collection S
-----> -----> -----> - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
AUDSYSAD                                     Z      N
AUDSYSAD                                     S      Y
AUDTABLE 'AUD%' DB2R5      T      A      N
AUDSYS   'SYSTABL SYSIBM  T      A      N
AUDEMP   'AUDEMP' DB2R5      T      A      N
AUDDEPT  'AUDDEPT DB2R5      T      A      N
AUDSECMA                                     A      N
AUDBADM                                     *      N
AUDBADM                                     *      N
AUDBADM                                     G      N
AUDBADM                                     *      N
AUDBADM                                     K      N
AUDVALID                                     A      N
AUDCUSTO 'CUSTOME DB2R5      T      A      N
```

Figure 13-63 Audit Policy list

From this panel, we see that there are already many policies set up. (For an explanation of these values, see Table 13-3.) To set up another policy, enter INS next to a line. This actions opens the Insert/Update Audit Policies panel (Figure 13-64).

```

ADBPZAPU ----- DB0B Insert/Update Audit Policies ----- 14:07
Command ==>

Enter Audit policy details:

Audit name . . .      > (? to lookup)
Object schema . .      (Optional)
Object name . . .      > (? to lookup)
Object type . . .      (A, C, P, T or blank)

Categories:
  Checking . . .      (A or blank)
  Validate . . .      (A or blank)
  Objmaint . . .      (A or blank)
  Execute . . .      (A, C or blank)
  Context . . .      (A or blank)
  Secmaint . . .      (A or blank)
  Sysadmin . . .      (I, L, O, R, S, * or blank)
  Dbadmin . . .      (B, C, D, E, G, K, M, P, T, * or blank)

DB name . . . . .      > (? to lookup)
Collection ID . .      > (? to lookup)
DB2 start . . . . .      (Yes/No)
  
```

Figure 13-64 Inserting an audit policy

The fields in the panel are explained in Table 13-3.

Table 13-3 Audit policy fields

Attribute name	Abbreviation on panel ADBPZAP	Value
Audit Name	NAME	Name of the Audit Policy.
Object Schema	Object Schema	Only applies to OBJMAINT and the Execute attribute.
Object Name	Object Name	Only applies to OBJMAINT and the Execute attribute.
Object Type	T	Only applies to OBJMAINT and Execute attribute <ul style="list-style-type: none"> ▶ A: Alias ▶ C: CLONE table ▶ P: Implicit tables created for XML columns ▶ T: Table
Checking	CH	The CHECKING category generates IFCID 140 trace records for authorization failures and IFCID 83 for authentication failures. <ul style="list-style-type: none"> ▶ A: Audit all failures ▶ Blank: Audit none

Attribute name	Abbreviation on panel ADBPZAP	Value
Validate	VA	<p>The VALIDATE category generates IFCID 55,83,87,169,312 trace records when there is an assignment or change of authorization ID and IFCID 269 when a trusted connection is established or used by a different user.</p> <ul style="list-style-type: none"> ▶ A: Audit All ▶ Blank: Audit none
Objmaint	OM	<p>The OBJMAINT category generates IFCID 142 trace records when the table identified in the Object name and the Object schema is altered or dropped.</p> <ul style="list-style-type: none"> ▶ A: Audit when table is altered/dropped ▶ Blank: Audit none
Execute	EX	<p>The EXECUTE category generates IFCID 143 and 144 records when the table identified in the Object name and the Object schema is accessed during the first operation performed by each unit of work (Option A). Also generates IFCID 145 records to trace bind time info about SQL statements that involve tables identified in the Object name and the Object schema (Option C).</p> <ul style="list-style-type: none"> ▶ Blank: No audit.
Context	CO	<p>The CONTEXT category generates IFCID 23,24 and 25 records to trace the start of utility, utility object, or phase change and the end of utility.</p> <ul style="list-style-type: none"> ▶ A: Audit all utilities ▶ Blank: Audit none
Secmaint	SM	<p>The SECMAINT category generates IFCID 141 trace records whenever a grant or revoke is made and an IFCID 270 trace record when a trusted context is created or altered.</p> <ul style="list-style-type: none"> ▶ A: Audit all ▶ Blank: Audit none
Sysadmin	SYSAD	<p>The SYSADMIN category generates IFCID 361 trace records whenever an operation is performed using an administrative authority to perform system administration tasks.</p> <ul style="list-style-type: none"> ▶ *: Audit all authorities ▶ I: Installation SYSADM ▶ L: SYSCTRL ▶ O: SYSOPR ▶ R: Installation SYSOPR ▶ S: SYSADM ▶ Blank: Audit none <p>You can concatenate supported characters, for example, LOS.</p>

Attribute name	Abbreviation on panel ADBPZAP	Value
Dbadmin	DBADM	<p>The DBADMIN category generates IFCID 361 trace records whenever an operation is performed using an administrative authority to perform database administration tasks.</p> <ul style="list-style-type: none"> ▶ *: Audit all authorities ▶ B: System DBADM ▶ C: DBCTRL ▶ D: DBADM ▶ E: SECADM ▶ B: System DBADM ▶ C: DBCTRL ▶ D: DBADM ▶ E: SECADM ▶ G: ACCESSCTRL ▶ K: SQLADM ▶ M: DBMAINT ▶ P: PACKADM ▶ T: DATAACCESS ▶ Blank: Audit none <p>You can concatenate supported characters, for example, BCD.</p>
DB Name	Database	The database name only applies to the DBADMIN category. It can be used to specify the database name for auditing DBADM, DBCTRL, and DBMAINT authorities.
Collection ID	Collection	The collection package name only applies to the DBADMIN category. It can be used to specify the collection name for auditing the PACKADM authority.
DB2 Start	DS	Indicates if audit policies are to be started automatically during DB2 start.

In our example, we set up two policies.

The first policy monitors a person with Database Administration authority on database ADMR3ADB. The first policy is called Policy1 and is defined in Figure 13-65.

```

ADBPZAPU ----- DB0B Insert/Update Audit Policies -----14:07
Command ==>

Enter Audit policy details:

Audit name . . . POLICY1 > (? to lookup)
Object schema . . . (Optional)
Object name . . . > (? to lookup)
Object type . . . (A, C, P, T or blank)

Categories:
  Checking . . . (A or blank)
  Validate . . . (A or blank)
  Objmaint . . . (A or blank)
  Execute . . . (A, C or blank)
  Context . . . (A or blank)
  Secmaint . . . (A or blank)
  Sysadmin . . . (I, L, O, R, S, * or blank)
  Dbadmin . . . D (B, C, D, E, G, K, M, P, T, * or blank)

DB name . . . . . ADMR3ADB > (? to lookup)
Collection ID . . . > (? to lookup)
DB2 start . . . . Y (Yes/No)

```

Figure 13-65 Define database auditing policy

Notice that we have defined this policy to start when DB2 starts. Alternatively, you can issue the start command shown in Example 13-22. The panels build the necessary SQL to populate the DB2 tables with the audit policy details.

Example 13-22 Starting trace with an audit policy

```
-START TRACE(AUDIT) AUDTPLCY(POLICY1) DEST(GTF)
```

The second policy monitors updates to the audit policies themselves. The policy records are stored in the SYSIBM.SYSAUDITPOLICY catalog table, so you need to audit the table. You need the EXECUTE parameter, as shown in Figure 13-66.

```
ADBPZAPU ----- DB0B Insert/Update Audit Policies -----14:07
Command ==>

Enter Audit policy details:

Audit name . . . POLICY2 > (? to lookup)
Object schema . . SYSIBM (Optional)
Object name . . . SYSAUDIT > (? to lookup)
Object type . . . T      (A, C, P, T or blank)

Categories:
  Checking . . .      (A or blank)
  Validate . . .      (A or blank)
  Objmaint . . .      (A or blank)
  Execute . . . C     (A, C or blank)
  Context . . .      (A or blank)
  Secmaint . . .      (A or blank)
  Sysadmin . . .      (I, L, O, R, S, * or blank)
  Dbadmin . . .      (B, C, D, E, G, K, M, P, T, * or blank)

DB name . . . . . > (? to lookup)
Collection ID . . > (? to lookup)
DB2 start . . . . yes (Yes/No)
```

Figure 13-66 Auditing changes to the audit tables

This policy audits the first access to this table during a unit of work or at bind time for SQL statements that reference this table in their DBRM. The Interpret command for this policy is shown in Figure 13-67.

```
ADBPZAPI ----- DBOB Interpretation of an Object in SYSAUDITPOLICIES -----14:07
Option ==>

Details for Audit Policy: POLICY2 MINFAIL

Object Schema   : SYSIBM
Object Name     . : SYSAUDITPOLICIES
Object Type     . : T - Table
Checking       . . . : blank - Audit none
Validate       . . . : blank - Audit none
Object Maint    . : blank - Audit none
Execute        . . . : C - Audit first insert/update/delete operation
Context        . . . : blank - Audit none
Security Maint  : blank - Audit none
System Admin   . : blank - Audit none
DB Admin       . . . : blank - Audit none
Database name   :
Collection ID   :
DB2 start      . . . : Y - Start automatically
Created TS     . . . : 2010-11-02-19.17.17.167563
Altered TS     . . . : 2010-11-02-19.17.17.167563
```

Figure 13-67 Interpret Policy 2

If you want to update the contents of a policy, issue the Update line command and the ADBPZAPU panel opens (Figure 13-66 on page 415).

Additionally, from the main panel, you can access the DB2 catalog details for objects that are part of an audit policy.



Physical design

In this chapter, we describe functions related to the table space physical design, that is, the function related to the definition of attributes for table spaces.

This chapter contains the following topics:

- ▶ INCLUDE column
- ▶ Inline LOBs
- ▶ XML
- ▶ Adding an active log data set
- ▶ ADD PARTITION
- ▶ TIMESTAMP with TIME ZONE
- ▶ ALTER BUFFERPOOL PAGESTEAL

14.1 INCLUDE column

DB2 10 for z/OS has the ability to add or include additional columns to a unique index. Some benefits of using an INCLUDE index are:

- ▶ Reduces the number of indexes defined on a table, which has a positive impact on maintenance.
- ▶ Improves performance through index only access.
- ▶ Fewer data sets to maintain.

More than one column may be included. The total length cannot exceed the size limit of DB2 itself.

14.1.1 Adding an INCLUDE column using the AL line command

Issue the AL line command next to the name of a unique index (Figure 14-1).

DB2 Admin ----- DSNT Indexes ----- Row 1 to 11 of 26

Command ==> Scroll ==> PAGE

Commands: DIS STA STO ALL

Line commands:

T - Tables D - Database G - Storage group P - Plans C - Columns

DIS - Display index space STA - Start index space STO - Stop index space

? - Show all line commands

Select	Index Name	Index Schema	Table Name	Table Schema	U	Cols	C	C	CC
	*	*	*	*	*	*	*	*	*
----->-----									
AL	TD87XA01_DEPT	TEAM87	TD87TB01_DEPT	TEAM87	P	1	Y	Y	YN
	TD87XB01_DEPT	TEAM87	TD87TB01_DEPT	TEAM87	D	1	N	N	YN
	TD87XC02_EMP	TEAM87	TD87TB02_EMP	TEAM87	P	1	N	N	YN
	TD87XB02_EMP	TEAM87	TD87TB02_EMP	TEAM87	D	1	N	N	YN
	TD87XA02_EMP	TEAM87	TD87TB02_EMP	TEAM87	U	3	Y	Y	YN
	TD87XA03_ACT	TEAM87	TD87TB03_ACT	TEAM87	P	1	Y	Y	YN
	TD87XA04_PROJECT	TEAM87	TD87TB04_PROJECT	TEAM87	P	1	Y	Y	YN
	TD87XA05_PROJACT	TEAM87	TD87TB05_PROJACT	TEAM87	P	2	Y	Y	YN
	TD87XB02_EPA	TEAM87	TD87TB06_EMPPROJAC	TEAM87	D	2	N	N	YN
	TD87XA01_EPA	TEAM87	TD87TB06_EMPPROJAC	TEAM87	P	3	Y	Y	YN
	TD87XA11_PARTTB	TEAM87	TD87TB11_PARTTB	TEAM87	P	3	Y	Y	YN

Figure 14-1 AL line command: Unique index

Click the ADDCOL keyword at the top of the panel or enter ADDCOL on the Command line (Figure 14-2).

```
DB2 Admin ----- DSNT Alter Index ----- Row 1 to 4 of 4
Command ==> addcol                               Scroll ==> PAGE

Commands: ADDCOL

ALTER INDEX  TEAM87.TD87XA02_EMP                (No. of Partitions: 4      )

Buffer Pool . . . . BP16      Close Rule . . . . YES  Copy Allowed . . NO
Piece Size . . . . .          Cluster . . . . . YES   Padded . . . . . NO
Compress . . . . . NO

Sel   Part      Pqty   Sqty  FreePg %Free Erase ST VCAT      Stogroup GBPCache
-----
All Parts:
      1          52     52     0    10 NO    I DSNTCAT  GLWG01  CHANGED
      2          52     52     0    10 NO    I DSNTCAT  GLWG01  CHANGED
      3          52     52     0    10 NO    I DSNTCAT  GLWG01  CHANGED
      4          52     52     0    10 NO    I DSNTCAT  GLWG01  CHANGED
***** END OF DB2 DATA *****
```

Figure 14-2 ADDCOL

Enter the I (Include) line command next to the column to be included (Figure 14-3).

```
DB2 Admin          DSNT Alter Index - Add Columns          Row 1 to 13 of 19
Command ==>                                              Scroll ==> PAGE

Line commands:  A - Add as ascending    D - Add as descending
                R - Add as random        I - Add as include
ALTER INDEX "TEAM87"."TD87XA02_EMP"
ADD COLUMN ( ... )

Sel Column Name      Col Type Length Null ColSeq Ord
*      *      *      *      *      *
-----
      LASTNAME      VARCHAR      N      1 A
      FIRSTNME      VARCHAR      N      2 A
      EMP_NO        INTEGER      N      3 A
      MIDINIT       CHAR          N
I  WORKDEPT        INTEGER      Y
      PHONENO       CHAR          N
      HIREDATE      DATE          N
      JOB           CHAR          Y
      MANAGER       CHAR          N
      EDLEVEL       SMALLINT      N
      SEX           CHAR          Y
      BIRTHDATE     DATE          N
      SALARY        DECIMAL      Y
```

Figure 14-3 Add as INCLUDE

An ALTER statement is generated (Figure 14-4).

```

DB2 Admin ----- DSNT Statement Execution Prompt ----- 13:03
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'TEAM76.ALTER.WSL'
Work statement list name ==> ALTER      Action ==> A (Append or Replace)
                                           More:      +

Statement that is about to be executed (first 28 lines):
ALTER INDEX "TEAM87"."TD87XA02_EMP"
ADD INCLUDE COLUMN ("WORKDEPT" )

```

Figure 14-4 ALTER statement of INCLUDE column

In this example, you execute the statement by entering a 1 on the Option line. As shown in Figure 14-5, WORKDEPT has a ColSeq of 4. Notice the SQL warn(+610) message in the upper right corner of the panel. This message indicates that the index is in the RBDP (Rebuild Pending) state.

```

DB2 Admin          DSNT Alter Index - Add Columns          SQL warn(+610)
Command ==>                                              Scroll ==> PAGE

Line commands:  A - Add as ascending   D - Add as descending
                R - Add as random       I - Add as include
ALTER INDEX "TEAM87"."TD87XA02_EMP"
ADD COLUMN ( ... )

Sel Column Name      Col Type Length Null ColSeq Ord
*                   * * * * *
-----
  LASTNAME           VARCHAR          N      1 A
  FIRSTNME           VARCHAR          N      2 A
  EMP_NO             INTEGER          N      3 A
  WORKDEPT           INTEGER          Y      4
  MIDINIT            CHAR              N
  PHONENO            CHAR              N
  HIREDATE            DATE              N
  JOB                 CHAR              Y
  MANAGER            CHAR              N
  EDLEVEL            SMALLINT          N
  SEX                 CHAR              Y
  BIRTHDATE           DATE              N
  SALARY             DECIMAL          Y

```

Figure 14-5 Column included: Results in a +610

After the ALTER command completes, the Alter Index Add Columns panel opens again, as other columns could be included in the index as well. In this example, you only added one column. Because the change is complete, press F3 a couple of times to get out of the change scenario. As a result, the Create Index - Utilities panel opens and provides the user with the opportunity to do a REBUILD and, potentially, a COPY and RUNSTATS as well (Figure 14-6).

```

DB2 Admin ----- DSNT Create Index - Utilities -- Row 1 to 3 of 3
Command ==>

Select the utilities to run for index TEAM87.TD87XA02_EMP
REBUILD ==> YES (Yes/No) RUNSTATS ==> (Yes/No)
COPY ==> (Yes/No)

Utility control options:
Review/modify options : NO (Yes/No)
Generate work statement list : NO (Yes/No)
Generate template statements : NO (Yes/No)
TU - Specify Template Usage

Name      Type Part  Status      PhyErrLo PhyErrHi Catalog Piece
-----
TD87XA02 IX    0001 RW,RBDP
      -THRU    0004
TD87XA02 IX    RW,RBDP
***** END OF DB2 DATA *****

```

Figure 14-6 Create index: Utilities

Select the utilities to run. Submit the generated utility job to complete the task.

14.1.2 Adding an INCLUDE column using the ALT line command

Issue the ALT line command next to the name of a unique index (Figure 14-7).

```

DB2 Admin ----- DSNT Indexes ----- Row 19 to 26 of 26
Command ==> Scroll ==> PAGE

Commands: DIS STA STO ALL
Line commands:
  T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
  DIS - Display index space  STA - Start index space  STO - Stop index space
  ? - Show all line commands

```

Select	Index Name	Index Schema	Table Name	Table Schema	U	Cols	C	C	CC
*	*	*	*	*	*	*	*	*	**
----->-----									
	TD89XA44_PNG	TEAM89	TD89TB44_PNG	TEAM89	U	1	Y	Y	YN
	TD89XA51_SQL	TEAM89	TD89TB51_SQL	TEAM89	U	2	Y	Y	YN
	TD89XA11_PARTTB	TEAM89	TD89TB11_PARTTB	TEAM89	P	3	Y	Y	YN
	TD89XB11_PARTTB	TEAM89	TD89TB11_PARTTB	TEAM89	D	2	N	N	YN
ALT	TD89XA02_EMP	TEAM89	TD89TB02_EMP	TEAM89	U	3	Y	Y	YN
	TD89XB02_EMP	TEAM89	TD89TB02_EMP	TEAM89	D	1	N	N	YN
	TD89XC02_EMP	TEAM89	TD89TB02_EMP	TEAM89	P	1	N	N	YN
	TD89XA61_SPL	TEAM89	TD89TB61_SPL	TEAM89	U	3	Y	Y	YN
***** END OF DB2 DATA *****									

Figure 14-7 Adding an INCLUDE column using the ALT line command

To include a column in the index, enter an I next to the column name (Figure 14-8).

```
DB2 Admin ----- DSNT Redefine Index ----- Row 1 from 19
Command ==>                                     Scroll ==> PAGE

Commands: CONTINUE ORIGINAL
Line commands: nnn A|D - Sequence & order R - Remove the column I - Include
A - Ascending D - Descending RA - Random U - Update expression/XMLpattern
B - Business Time without overlaps

CREATE INDEX TEAM89 . TD89XA02_EMP >
ON TEAM89.TD89TB02_EMP
Unique . . . . . YES Where Not Null . . . Cluster . . . . . YES
Buffer Pool . . . . BP16 Close Rule . . . . YES Copy Allowed . . NO
Define . . . . . YES Defer . . . . .
Partitioned . . . . YES Padded . . . . . NO Compress . . . . NO

Select Column Name Col Type Length Scale N ColSeq Ord OldSeq Ord
* * * * *
-----
LASTNAME VARCHAR 20 0 N 1 A 1 A
FIRSTNME VARCHAR 12 0 N 2 A 2 A
EMP_NO INTEGER 4 0 N 3 A 3 A
MIDINIT CHAR 1 0 N
i WORKDEPT INTEGER 4 0 Y
PHONENO CHAR 4 0 N
```

Figure 14-8 INCLUDE column

Press Enter, and the WORKDEPT column moves up and is placed in sequential order with the other columns in the index (Figure 14-9).

```

DB2 Admin ----- DSNT Redefine Index ----- Row 1 from 19
Command ==>                                     Scroll ==> PAGE

Commands: CONTINUE ORIGINAL
Line commands: nnn A|D - Sequence & order R - Remove the column I - Include
A - Ascending D - Descending RA - Random U - Update expression/XMLpattern
B - Business Time without overlaps

CREATE INDEX TEAM89      . TD89XA02_EMP      >
      ON TEAM89.TD89TB02_EMP
Unique . . . . . YES      Where Not Null . . .      Cluster . . . . . YES
Buffer Pool . . . . BP16      Close Rule . . . . . YES      Copy Allowed . . NO
      Define . . . . . YES      Defer . . . . .
Partitioned . . . . YES      Padded . . . . . NO      Compress . . . . NO

Select Column Name      Col Type      Length      Scale N      ColSeq Ord OldSeq Ord
      *              *              *              * *              * *              * *
-----
      LASTNAME      VARCHAR      20      0 N      1 A      1 A
      FIRSTNME      VARCHAR      12      0 N      2 A      2 A
      EMP_NO      INTEGER      4      0 N      3 A      3 A
*      WORKDEPT      INTEGER      4      0 Y      30 I
      MIDINIT      CHAR      1      0 N
      PHONENO      CHAR      4      0 N

```

Figure 14-9 INCLUDE column using ALT

Note: When using AL to include a column in a unique index, the sequence number of the included column is incremented by 1. When using ALT, the sequence number of the included column is 30.

To make the change, click the CONTINUE keyword at the top of the panel or enter CONTINUE on the Command line and press Enter (Figure 14-10).

```
DB2 Admin ----- DSNT Redefine Index ----- Row 1 from 19
Command ==> continue                               Scroll ==> PAGE

Commands: CONTINUE ORIGINAL
Line commands: nnn A|D - Sequence & order R - Remove the column I - Include
A - Ascending D - Descending RA - Random U - Update expression/XML
pattern
B - Business Time without overlaps

CREATE INDEX TEAM89      . TD89XA02_EMP      >
      ON TEAM89.TD89TB02_EMP
Unique . . . . . YES      Where Not Null . . .      Cluster . . . . . YES
Buffer Pool . . . . BP16      Close Rule . . . . . YES      Copy Allowed . . NO
      Define . . . . . YES      Defer . . . . .
Partitioned . . . . YES      Padded . . . . . NO      Compress . . . . NO

Select Column Name      Col Type      Length      Scale N      ColSeq Ord      OldSeq Ord
      *              *              *              * *              * *              * *
-----
      LASTNAME      VARCHAR      20      0 N      1 A      1 A
      FIRSTNME      VARCHAR      12      0 N      2 A      2 A
      EMP_NO      INTEGER      4      0 N      3 A      3 A
*      WORKDEPT      INTEGER      4      0 Y      30 I
      MIDINIT      CHAR      1      0 N
      PHONENO      CHAR      4      0 N
```

Figure 14-10 ALT INCLUDE column

The index you are working with is a CLUSTERING index, so the panel shown in Figure 14-11 opens.

```
DB2 Admin -- DSNT Clustering Index Confirmation ----- 14:

A clustering index already exists for the table.
Specify whether to continue or return.

Select a choice
1 1. Continue specification as a clustering index.
2. Continue, but remove clustering index clause.
3. Display the current clustering index.
4. Return
```

Figure 14-11 Clustering Index Confirmation panel

In this example, select “1. Continue specification as a clustering index”. The Redefine Index panel opens (Figure 14-12).

```

DB2 Admin ----- DSNT Redefine Index - Space ----- Row 1 to 4 of 4
Command ==> continue                               Scroll ==> PAGE

Commands: CONTINUE ORIGINAL
Line commands: 0 - Original data  C - Clear data

CREATE INDEX TEAM89.TD89XA02_EMP
              ON TEAM89.TD89TB02_EMP
Sel   Part      Pqty   Sqty FreePg %Free Erase ST VCAT      Stogroup GBPCache
      *          *      *      *      *  *  *  *  *      *      *
-----
Default :          52    52     0    10          I DSNTCAT  GLWG01  CHANGED
      1
      2
      3
      4
***** END OF DB2 DATA *****

```

Figure 14-12 Redefine Index

The changes have all been specified, so click the CONTINUE keyword at the top of the panel or enter CONTINUE on the Command line. The DDL for the index opens (Figure 14-13).

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDIT      SYS10316.T153304.RA000.DBA104.R0109016          Columns 00001 00072
Command ==>                               Scroll ==> CSR
***** ***** Top of Data *****

000001  CREATE UNIQUE INDEX
000002  "TEAM89"."TD89XA02_EMP"
000003  ON
000004  "TEAM89"."TD89TB02_EMP"
000005  ( "LASTNAME" ASC,
000006  "FIRSTNME" ASC,
000007  "EMP_NO" ASC)
000008  INCLUDE( "WORKDEPT")
000009  USING STOGROUP GLWG01
000010  PRIQTY 52
000011  SECQTY 52
000012  FREEPAGE 0
000013  PCTFREE 10
000014  GBPCACHE CHANGED
000015  DEFINE YES
000016  CLUSTER
000017  PARTITIONED
000018  NOT PADDED
000019  BUFFERPOOL BP16

```

Figure 14-13 Index DDL

Press F3 to open the Alter Tables panel. More changes could be defined on this panel using the center of the panel to open a list of tables. Because our changes have all been defined, click the ALTER keyword at the top of the panel or enter ALTER on the Command line (Figure 14-14).

```
DB2 Admin ----- DSNT Alter Tables ----- Row 1 to 1 of 1
Command ==> alter                                     Scroll ==> PAGE

Commands:
  ALTER - generate jobs  ADDFK - Add Foreign Key-affected tables
  OPTIONS - Change alter options
Line commands:
  A - Alter Object  D - Delete  S - Select Object  REL - Alter related
  FK - Add Foreign Key-affected tables  E - Edit DDL
  RS - Reset RI-FK flags

To add a table, provide a schema and name below and hit Enter
Schema . . . > (Table Schema)
Name . . . . > (Table Name. ? to look up)

      Object  Object
Sel Qual  Name          T DB Name  TS Name  RI RI  FK  Chg
      *      *          * *      *      Rels Add Add Rqd Oper
                        * *      *      *      *      *
-----
      TEAM89  TD89XA02_EMP  X          NA  NA      MODIFY
*****
                                END OF DB2 DATA *****
```

Figure 14-14 Alter Tables panel

The Build Analyze and Apply Job panel opens (Figure 14-15).

```
----- ALTER - Build Analyze and Apply Job -----
Option ==>

Specify the following:

Worklist information:
  Worklist name . . . . . ALTER    (also used as middle qualifier in
  DSNs)
  Prefix for data sets . . . TEAM76

Data set information:
  PDS final qualifiers . . . JCL.CNTL
  Member name . . . . . ADBALTER
  Delete member name . . . ADBDELET (Optional job to delete work data
  sets)

Options:
  Generate online . . . . . NO      (Yes/No)
  Generate one job . . . . . YES    (Yes/No)
  Member name or prefix . . APPLY
  As work statement list . . YES    (Yes/No)
  Unload method . . . . . U        (Unload, Parallel unload, HPU)

Optional processes:
  Run CHECK DATA . . . . . NO      (Yes/No)
```

Figure 14-15 Build Analyze and Apply panel

Enter or edit any necessary parameters on the panel and press Enter. The Data Set Existence Check panel might open. Selected option 1. Continue by having the data sets deleted and recreated, which means that old data sets are dropped and recreated for this execution. The other option is to specify a new qualifier so that different data sets can be allocated (Figure 14-16).

```
DB2 Admin ----- Data set Existence Check -----

Data sets for the chosen prefix and worklist already exist.
Continuing will cause the loss of all existing content in those data sets.

Select a choice:
  1 - Continue by having the data sets deleted and recreated.
  2 - Use a new worklist qualifier . .

The current data set qualifiers are: TEAM76.ALTER
```

Figure 14-16 Data set Existence Check panel

You are prompted to specify the work statement list library name (Figure 14-17).

Note: At the time of the writing of this book, the work statement list library name is generated. The format for the data set name is your user ID, the work list member name as specified on the Build Analyze and Apply Job panel, and a constant of WSL. You need to be careful here because each time you change the work list name on the parameter panel a new work statement list library is created.

```
DB2 Admin ----- Specify Work Statement List -----  
  
Work stmt list dsn . . . 'TEAM76.WSL.library'  
Work stmt list name . . . includ02
```

Figure 14-17 Work Statement List library

A batch JCL job stream, with a default name of ADBALTER, is generated in the PDS specified on the Build Analyze and Apply Job panel. Submit the job to create the WSL member in the work statement list library. You should get a MAXCC = 0 statement in return. After the job is complete, issue the WSL command on the Command line to view the WSL member.

After the WSL library is displayed, there are a number of options to choose from to view the member:

► Interpret

This option displays the Interpret Work Statement List Options panel. Press Enter and a condensed list of actions is displayed. For the purposes of this exercise, you should only see an ALTER. The actual ALTER statement may be displayed by using the V -Views line command.

► Show

A panel is displayed showing the type of statements making up the WSL member and the control card statements.

► Edit

This option allows you to view the statements in full screen edit. TSO edit commands are supported.

[illegible]

When run online, the 610 message is displayed at the end of the run (Figure 14-19).

[illegible]

Figure 14-19 Online WSL execution output

In both cases, the user has to assume the responsibility of performing a REBUILD index independent from the product.

14.1.3 Removing an INCLUDE column

DB2 does not provide an SQL syntax or an ALTER option for removing an INCLUDE column. Removal requires the index to be dropped and created. Consequently, you cannot use the DB2 Administration Tool AL command to remove an INCLUDE column, but you can use the R (Remove column) option of ALT (Figure 14-20).

```
DB2 Admin ----- DSNT Redefine Index ----- Row 1 from 19
Command ==>                                     Scroll ==> PAGE

Commands: CONTINUE ORIGINAL
Line commands: nnn A|D - Sequence & order  R - Remove the column I - Include
A - Ascending D - Descending RA - Random U - Update expression/XMLpattern
B - Business Time without overlaps

CREATE INDEX TEAM89      . TD89XA02_EMP      >
      ON TEAM89.TD89TB02_EMP
Unique . . . . . YES      Where Not Null . . .      Cluster . . . . . YES
Buffer Pool . . . . BP16      Close Rule . . . . . YES      Copy Allowed . . NO
      Define . . . . . YES      Defer . . . . .
Partitioned . . . . YES      Padded . . . . . NO      Compress . . . . NO

Select Column Name      Col Type      Length      Scale N ColSeq Ord OldSeq Ord
      *                  *                  *          * *      * *      * *
-----
      LASTNAME      VARCHAR      20          0 N      1 A      1 A
      FIRSTNME      VARCHAR      12          0 N      2 A      2 A
      EMP_NO      INTEGER      4          0 N      3 A      3 A
r      WORKDEPT      INTEGER      4          0 Y      4          4
      MIDINIT      CHAR      1          0 N
      PHONENO      CHAR      4          0 N
```

Figure 14-20 Removing an INCLUDE column

After completing the process, the generated SQL no longer contains the `INCLUDE` column (Figure 14-21).

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
```

```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS  
EDIT          SYS10316.T165416.RA000.DBA104.R0109339           Columns 00001 00072  
Command ==>                                           Scroll ==> CSR  
***** ***** Top of Data *****  
000001 CREATE UNIQUE INDEX  
000002 "TEAM89"."TD89XA02_EMP"  
000003 ON  
000004 "TEAM89"."TD89TB02_EMP"  
000005 ( "LASTNAME" ASC,  
000006    "FIRSTNME" ASC,  
000007    "EMP_NO" ASC)  
000008 USING STOGROUP GLWG01  
000009     PRIQTY 52  
000010     SECQTY 52  
000011 FREEPAGE 0  
000012 PCTFREE 10  
000013 GBPCACHE CHANGED  
000014 DEFINE YES  
000015 CLUSTER  
000016 PARTITIONED  
000017 NOT PADDED  
000018 BUFFERPOOL BP16  
000019 CLOSE YES
```

Figure 14-21 Generated SQL syntax for the index

DB2 Administration Tool generates the appropriate JCL to complete the change.

14.2 Inline LOBs

This section looks at DB2 Administration Tool V10 support for inline LOBs.

14.2.1 What are Large Objects

Large Objects (LOBs) are used to store lengthy documents, such as resumes, the text of a novel, or graphics images, such as pictures or objects that need to be stored in their native format.

LOBs are defined as columns on a table and consist of three types:

- ▶ Binary Large Objects (BLOBs)
- ▶ Character Large Objects (CLOBs)
- ▶ Double-byte Character Large Objects (DBCLOBs)

BLOBs are used to store non-traditional types of data or objects where the native formats must be preserved, such as .JPEG and .GIF files. CLOBs are used to store lengthy documents containing characters that can be stored using UTF-8 (or single-byte) format.

DBCLOBs are also used to store lengthy documents, but store those that contain characters in a UTF-16 (or double-byte) character set.

14.2.2 What are inline LOBs

With the release of DB2 10 for z/OS, you can now specify an amount of the LOB data to be stored in the base table, as opposed to all the data being stored in the auxiliary table. If the LOB length is longer than the specified inline length, the remainder of the LOB data is stored on the auxiliary table. If the LOB is small enough, this can result in the entire LOB being stored on the base table, which reduces the processing required to return LOB data to the application.

The amount of LOB data that is stored in the base table can be controlled two ways:

- By setting a system default in DSNZPARM, using the `INLINE_LOB_LENGTH` parameter, this sets the amount of a LOB that will be stored in the base.
- By specifying the amount of inline data when creating or altering the table.

DB2 Administration Tool can assist with the defining of the amount of inline LOB data and generating the necessary steps to implement the change.

14.2.3 Creating a table with an inline LOB column

To create a table with an inline lob column, define the table as you would normally with an external LOB. Start by either issuing the CRE line command against a table or CT from the CREATE/DROP panel (enter DDL from the System Catalog panel, or option 7.2 from the main panel).

After entering the name of the table and the number of columns, the Create Table Columns panel (ADB26CTF) opens. Complete the necessary fields (Figure 14-22). You are using `CURRENT RULES = 'DB2'` and have specified a ROWID Column for the LOB allowing you to follow any site naming standards that may exist.

ADB26CTF ----- VA1A Create Table Columns -----
Row 1 to 2 of 2

Command ==>
Scroll ==> PAGE

Schema . . ADMR3 >
Database . . .

Name . . . T99_LOBTABLE >
Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
 Line commands: M - Move A - After B - Before
 Inn - Insert U - Update D - Delete Rnn - Repeat

Select	Column Name	Col Type	Length	Scale	Null	D	Col	No	Type	Operation
*	*	*	*	*	*	*	*	*	*	*
-----> -----										
*	LOBCOL	CLOB	386640		0 N	Y		1		INSERT
*	T99_ROWID	ROWID	17		0 N	A		2		INSERT
***** END OF DB2 DATA *****										

Figure 14-22 Creating a LOB column

The LOB defined in Figure 14-22 on page 434 is currently either a regular LOB (all in the AUX table space) or it will be an inline LOB with an inline length equivalent to the default supplied by the DSNZPARM, `INLINE_LOB_LENGTH`.

To make the LOB inline, or to change the length of the inline portion of the LOB column, issue a “U” against the column (Figure 14-23).

```

ADB26CTF ----- VA1A Create Table Columns ----- Row 1 to 2 of 2
Command ==>                                           Scroll ==> PAGE

Schema . . ADMR3      >          Database . . .
Name . . . T99_LOBTABLE      > Table space . .

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands:  M - Move  A - After  B - Before
               Inn - Insert U - Update D - Delete Rnn - Repeat

Select Column Name      Col Type  Length Scale  Null D Col No Type      Operation
      *                  *          *         * *    *      * *
-----> -----
U      LOBCOL            CLOB      386640      0 N    Y        1 INSERT
*      T99_ROWID         ROWID      17         0 N    A        2 INSERT
***** END OF DB2 DATA *****

```

Figure 14-23 Creating an inline LOB

This action opens the Create Table Column Number panel, where you can change the inline length. The inline length field is set to either 0 or the default (Figure 14-24). Overwrite this value with the desired value and continue with the creation process.

```

ADB26CUU ----- VA1A Create Table Column Number      1 ----- 11:08
Command ==>

CREATE TABLE                                     Schema . . . ADMR3      >
                                                    Name . . . . LOBTABLE      >
                                                    More:      +
Column name . . . LOBCOL      > (Column number      1)
Data type . . . . CLOB      (Built-in only)
Data length . . . 386640      (Built-in only)
INLINE LENGTH . . 0      (0-32680 BLOB or CLOB, 0-16340 DBCLOB)
Precision . . . .      (FLOAT and DECIMAL only)
Scale . . . . .      (DECIMAL and timestamp types only)
Type schema . . .      > (User-defined type schema)
Type name . . . .      > (? to look up)
WITH TIME ZONE .      (Yes/No - for TIMESTAMP only)

Allow Nulls . . . NO      (Yes or blank--nullable, No-NOT NULL)
FOR ? DATA . . .      (B-Bit, S-SBCS, M-Mixed, blank-N/A)
WITH DEFAULT . . YES      (Yes, No, L (SECLABEL) or enter value below)
Default value . .
GENERATED . . . .      (A-ALWAYS,      D-DEFAULT,
                        I-ALWAYS AS IDENTITY,      J-DEFAULT AS IDENTITY,
                        E-ALWAYS AS UPD TIMESTAMP,      F-DEFAULT AS UPD TIMESTAMP,
                        Q-ALWAYS AS ROW BEGIN,      R-ALWAYS AS ROW END,

```

Figure 14-24 Updating inline length

This action results in the inline LOB length being set to the desired value.

14.2.4 Determining the length of the inline portion of the LOB column

If you want to determine the inline portion of the LOB column, you can either generate the DDL for the table or you can display the columns of the table by using the C line command against the table (Figure 14-25).

```
ADB21TC n -- VA1A Columns in Table ADMR3.T99SPUFILO > ----- Row 1 to 2 of 2
Command ==>                                           Scroll ==> PAGE

Line commands:
T - Tables  X - Indexes  A - Auth  GR - Grant  H - Homonyms  I - Interpret
UR - Update runstats  LAB - Label  COM - Comment  DI - Distribution stats
? - Show all line commands

Select Column Name          Col No Col Type Length Scale  Null Def FP      Col Card
      *                  * *          *      * *      * *
----->-----
      LOBCOL                1 CLOB          103      0 N    Y  N      -1
      DB2_GENERATED_ROWID    2 ROWID          17      0 N    A  N      -1
***** END OF DB2 DATA *****
```

Figure 14-25 Inline LOB size

The length of the column is equal to the inline length plus the additional 4 byte pointer, so the inline portion of the LOB is 99 bytes.

14.2.5 Altering the inline length of the LOB

There are times when you must alter the inline length of the LOB, such as when the profile of the data has changed and it would be beneficial to have more, or less, data stored within the table to optimize performance. DB2 Administration Tool can help you implement the change by using the ALT command, and DB2 Administration Tool provides the JCL to complete the transition, including all necessary utilities.

To start the alter use the ALT command against the table that contains the columns that you want to change, in this case, ADMR3.T99_LOBTABLE (Figure 14-26).

```
ADB21T in ----- VA1A Tables, Views, and Aliases ----- Row 1 to 4 of 4
Command ==> Scroll ==> CSR

Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands
```

Sel	Name	Schema	T DB Name	TS Name	Cols	Rows	Chks	C
*	*	*	* *	*	*	*	* *	*
ALT	T99_LOBTABLE	ADMR3	T DSN00010	T99RLOBT	2	-1	0	
	T99_LLOBCODT9Z26I7	ADMR3	X DSN00010	LDT9ZIT7	3	-1	0	
	T99SPUFILOB	ADMR3	T DSN00011	T99SPUFI	2	-1	0	
	T99SPLOBCODT90PZLG	ADMR3	X DSN00011	LDT9Z8XG	3	-1	0	

```
*****
END OF DB2 DATA *****
```

Figure 14-26 Selecting a table to alter

Press Enter. The ALTER table panel opens and shows the table and column details. Select the column you want to change by using the U line command against it (Figure 14-27).

```
ADB27C in ----- VA1A ALTER Table ----- Row 1 to 2 of 2
Command ==> Scroll ==> PAGE

New schema . . ADMR3 > Old schema: ADMR3
New name . . . T99_LOBTABLE > Old name : T99_LOBTABLE
Volatile . . . NO New DB . . DSN00010
Rows per page: 123 Partitions: 1 New TS . . T99RLOBT

Commands: CONTINUE PRIMKEY ADD REL ADDPART
Line commands:
I - Insert U - Update D - Delete R - Repeat LAB - Label COM - Comment
M - Move A - After B - Before X - Index RES - Reset update
Old Operation
```

Sel	Column Name	Col No	Col Type	Length	Scale	N D	Col No	Type
*	*	* *	*	*	*	* * *	* *	*
U	LOBCOL	1	CLOB	1048576	0	N Y	1	
	T99_ROWID	2	ROWID	17	0	N A	2	

```
*****
END OF DB2 DATA *****
```

Figure 14-27 Selecting a column to alter

This action opens the ALTER Table panel (Figure 14-28).

```

ADB26CTU ----- VA1A ALTER Table ----- 15:52
Command ==>

DB2 Admin ALTER                               Schema . . : ADMR3      >
                                                Name . . : T99_LOBTABLE  >
Column name . . LOBCOL                       > (column number 1)
Column type . . CLOB                         (CHAR,DECIMAL,INTEGER,SMALLINT,etc.)
Data length . . 1048576
Inline length . 99                          (0-32680 BLOB or CLOB, 0-16340 DBCLOB)
Precision . . .                             (FLOAT and DECIMAL only)
Scale . . . . .                             (DECIMAL and timestamp types only)
Type schema . .                             (User-defined type schema)
Type name . . .                             (User-defined type name)
WITH TIME ZONE .                            (Yes/No - for TIMESTAMP only)

Allow Nulls . . NO   (Yes-Nullable, No-NOT NULL)
FOR ? DATA . . .   (B - Bit, S - SBCS, M - Mixed, or blank)
WITH DEFAULT . . YES (Yes, No, L (SECLABEL) or enter value below)
Default value .

GENERATED . . .     (A-ALWAYS, D-DFLT, I-ALWAYS AS IDENT, J-DFLT AS IDENT,
                    E-ALWAYS AS UPD TIMESTAMP, F-DFLT AS UPD TIMESTAMP)
FIELDPROC

```

Figure 14-28 Altering inline length

As you can see, the inline length can be changed in this panel. In this example, change the length to 50. Press Enter and you return to the previous panel, where you can see an update marker showing against the LOB column (Figure 14-29).

```

ADB27C in ----- VA1A ALTER Table ----- Row 1 to 2 of 2
Command ==>                                     Scroll ==> PAGE

New schema . . ADMR3      >                      Old schema: ADMR3
New name . . . T99_LOBTABLE      >                Old name  : T99_LOBTABLE
Volatile . . . NO                      New DB   . . DSN00010
Rows per page: 14          Partitions: 1          New TS   . . T99RLOBT

Commands: CONTINUE PRIMKEY ADD REL  ADDPART
Line commands:
  I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
  M - Move    A - After   B - Before  X - Index   RES - Reset update
                                           Old Operation
Sel Column Name          Col No Col Type      Length  Scale N D Col No Type
      *                  * *
----->-----
*  LOBCOL                1 CLOB              1048576      0 N Y      1 UPDATE
  T99_ROWID              2 ROWID               17          0 N A      2
***** END OF DB2 DATA *****

```

Figure 14-29 Column update marker

Select continue and the ALTER Analysis Options panel opens (Figure 14-30).

```

ADBP7P in ----- VA1A ALTER Analysis Options ----- 16:04
Option ==>

Please specify the following for DB2 Admin ALTER:

Analysis options:
  Run SQLID . . . . . (Blank, an SQLID, or <NONE>)
  Use DEFER YES . . . . . NO (Yes/No)
  VIEW Column List . . . . . YES (Yes/No)
  Perform recovery analysis . . . . . YES (Yes/No)
  Enable authorization switching . . . NO (Yes/No)

  Perform analysis in batch . . . . . YES (Yes/No)

  Show this panel prior to each use . . . YES (Yes/No)

  Change diagnostic options . . . . . NO (Yes/No)

```

Figure 14-30 ALTER Analysis Options panel

This panel might not open if you have turned the prompt off. For more information about this panel and the options listed on it, see Chapter 7, “The ALT line command” on page 165.

Press Enter. The Build Analyze and Apply Job panel opens (Figure 14-31). Again, refer to Chapter 7, “The ALT line command” on page 165 for details about using this panel.

```

ADBPALT ----- ALTER - Build Analyze and Apply Job -----
Option ==>

Specify the following:

Worklist information:
  Worklist name . . . . . ALTLOB  (also used as middle qualifier in DSNs)
  Prefix for data sets . . . SYSADM

Data set information:
  PDS final qualifiers . . . JCL.CONTROL
  Member name . . . . . ADBALTER
  Delete member name . . . ADBDELET (Optional job to delete work data sets)

Options:
  Generate online . . . . . NO      (Yes/No)
  Generate one job . . . . . YES     (Yes/No)
  Member name or prefix . . . . . ALTLB
  As work statement list . . . YES   (Yes/No)
  Unload method . . . . . U         (Unload, Parallel unload, HPU)
  Authorization Switch ID . . .      (SQLID to sign on as, blank or <NONE>)
  SECADM Authorization ID . . .      (An ID to sign on as, blank or <NONE>)

Optional processes:

```

Figure 14-31 Build Analyze and Apply Job panel

We create a work statement list (WSL) named LOBALT that contains the steps needed to make this change. For more details about WSL, refer to Chapter 10, “Work statement lists” on page 259.

To make sure that all necessary utilities are generated to implement the change without leaving the object in a restrictive state, ensure that the RUN REORG/REBUILD flag is *not* set to N, as this setting suppresses the generation.

Generate a WSL (Figure 14-31). The WSL contain all the statements needed to make this change. The WSL is shown in Example 14-1.

Example 14-1 Generated WSL

```

-- Created by SYSADM on 2010/11/08 at 11:02
-- Generated by apply exec by SYSADM on 2010/11/08 at 11:02
--#ADMIN PROCESS CREATE
  ALTER TABLE ADMR3.T99_LOBTABLE
    ALTER COLUMN LOBCOL
      SET INLINE LENGTH 50 ;
  COMMIT ;
--#RESTART 1
PARALLEL REORG;
JOB;
TSODELETE 'SYSADM.VA1A.ALTLOB.CNT.S0001';
TSODELETE 'SYSADM.VA1A.ALTLOB.OUTPUT.S0001';

```

```

TSODELETE 'SYSADM.VA1A.ALTLOB.SDISC.S0001';
TEMPLATE SYSCOPY DSN 'SYSADM.VA1A.IC.S0001(+1)'
    UNIT SYSDA;
TEMPLATE UTLPUNCH DSN 'SYSADM.VA1A.ALTLOB.CNT.S0001'
    UNIT SYSDA;
TEMPLATE UTLREC DSN 'SYSADM.VA1A.ALTLOB.OUTPUT.S0001'
    UNIT SYSDA;
TEMPLATE UTLDISC DSN 'SYSADM.VA1A.ALTLOB.SDISC.S0001'
    UNIT SYSDA;
TEMPLATE UTLUT1 DSN 'SYSADM.VA1A.ALTLOB.SUT1.S0001'
    UNIT SYSDA;
TEMPLATE SORTOUT DSN 'SYSADM.VA1A.ALTLOB.SORTOUT.S0001'
    UNIT SYSDA;
LISTDEF ADBLDO INCLUDE TABLESPACES ALL
TABLE "ADMR3"."T99_LOBTABLE";
--#RESTART 2
REORG TABLESPACE LIST ADBLDO
    LOG NO
    SORTDEVT SYSDA
    SORTNUM 4
    COPYDDN(SYSCOPY)
    SHRLEVEL REFERENCE
    PUNCHDDN UTLPUNCH
    DISCARDN(UTLDISC)
    UNLDDN UTLREC
    WORKDDN(UTLUT1,SORTOUT);
--#RESTART 3
ENDJOB;
ENDPARALLEL;
-- End of Apply statements

```

Notice that a REORG has been generated for the table space. If you had said NO to the REORG or if you had increased the length of the column, then the WSL would have just contained an ALTER statement.

14.3 XML

In this section, we discuss DB2 Administration Tool V10 support for XML objects. We discuss creating or altering XML columns and indexes to make use of new features within DB2 10 for z/OS:

- ▶ XML Schema validation
- ▶ XMLPATTERN index support

14.3.1 What is XML

XML technology has become pervasive in virtually all industries and sectors due to its versatility and neutrality for exchanging data among diverse devices, applications, and systems from different vendors. These qualities of XML, along with its easy to understand self-describing nature, ability to handle structured, semi-structured, and unstructured data, and support for Unicode, have made XML a universal standard for data interchange.

DB2 9 for z/OS introduced support for XML and DB2 10 for z/OS has extended this support. DB2 Administration Tool assists you with the defining of tables and indexes containing the XML constructs.

DB2 Administration Tool support of XML primarily focuses on creating the XML attributes and navigating through the catalog.

14.3.2 Creating a table with XML columns

To create a table with an XML column, define the table as you would define a non-XML table. Issue CRETAB against the table space and select a table name. The Create Table Columns panel (ADB26CTF) opens. We enter the column details for the table (Figure 14-32).

```
ADB26CTF ----- DB0B Create Table Columns ----- Row 1 to 3 of 3
Command ==> Scroll ==> PAGE

Schema . . ADMR3      >      Database . . . ADMR3ADB
Name . . . BK_TO_CSTMRTMT  > Table space . . TSAUDIT1

Commands : CREATE PRIMKEY TBLOPTS PART
Line commands: M - Move A - After B - Before
Inn - Insert U - Update D - Delete Rnn - Repeat

Select Column Name      Col Type Length Scale Null D Col No Type
      *                  *      *      * *      *      * *
-----> -----
      MSG_ID             VARCHAR      35      0 N      N      1 UPDATE
      MSG_CRE_DT_TM      TIMESTMP      7      0 N      N      2 UPDATE
U      BK_TO_CSTMRTMT    XML           6      0 N      N      3 UPDATE
***** END OF DB2 DATA*****
```

Figure 14-32 Creating an XML table

Enter a U next to the XML column. The Create Table Column Number (ADB26CUU) panel opens (Figure 14-33).

ADB26CUU

----- DBOB Create Table Column Number

3

----- 17:36

Command ==>

CREATE TABLE

Column name . . . BK_TO_CSTMRT_STMT

Data type XML

Data length . . .

Precision

Scale

Schema name . . .

Type name

Allow Nulls . . . NO

FOR ? DATA . . .

WITH DEFAULT . . NO

Default value . .

GENERATED

FIELDPROC

Program name . .

Schema . . . ADMR3

Name BK_TO_CSTMRT_STMT

(Column number 3)

(Built-in only)

(Built-in only)

(Built-in only)

(Built-in only)

> (User-defined only)

> (? to look up)

(Yes-Nullable, No-NOT NULL)

(B-Bit, S-SBCS, M-Mixed, blank-N/A)

(Yes, No, L (SECLABEL) or enter value below)

(A-ALWAYS, D-DFLT, I-ALWAYS AS IDENT, J-DFLT AS IDENT, E-ALWAYS AS UPD TIMESTAMP, F-DFLT AS UPD TIMESTAMP)

(Optional)

More: +

>

>

Figure 14-33 Column details

The instructions to add the XML Modifier are the same as shown in 14.3.3, “Altering a table to add an XML column with an XML modifier” on page 444.

14.3.3 Altering a table to add an XML column with an XML modifier

In this section, you modify the table that we created to add an XML column. You add the BK_TO_CSTMRT_STMT_OLD column to the table, which adds an XML modifier to the column.

For more details about the XML functions within DB2 10 for z/OS, see *Extremely pureXML in DB2 10 for z/OS*, SG24-7915.

List the table to which you want to add the column by opening the Tables, Views, and Aliases panel (Figure 14-34).

```
ADB21T in ----- DBOB Tables, Views, and Aliases ----- Row 1 from 1
Command ==> Scroll ==> PAGE

Commands: GRANT MIG ALL
Line commands:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands

Sel  Name                Schema  T DB Name TS Name  Cols      Rows Chks C
      *                  *      * *      *      *          *   *  *
-----
      BK_TO_CSTMRT_STMT  ADMR3   T ADMR3ADB TSAUDIT1    2        -1    0
*****
***** END OF DB2 DATA *****
```

Figure 14-34 Adding XML to a table

Use the AL command to add a column. The Alter Table panel opens (Figure 14-35).

```
ADB21TA n ----- DBOB Alter Table ----- 18:14
Command ==>

Table schema . . : ADMR3    >
Table name . . . : BK_TO_CSTMRT_STMT  >

AUDIT . . . . . NONE      (None, Changes, or All)
DATA CAPTURE . . . . . NONE (None/Changes)
VALIDPROC . . . . . NULL   (NULL/Program name)
RESTRICT ON DROP . . . NO   (Yes/No)
VOLATILE . . . . . NO      (Yes/No)
APPEND . . . . . NO        (Yes/No)

ALTER TABLE with any of the above changes OR select one of the options below

More:  +

ADD column                ADD MATERIALIZED QUERY
ADD PRIMARY KEY           DROP MATERIALIZED QUERY
DROP PRIMARY KEY          REFRESH MATERIALIZED TABLE
ADD FOREIGN KEY           ADD PARTITIONING KEY
DROP FOREIGN KEY          ADD PARTITION
ADD CHECK constraint      ADD CLONE
DROP CHECK constraint     DROP CLONE
ADD UNIQUE constraint     ADD VERSIONING
```

Figure 14-35 Adding XML to a table: AL

Select the ADD Column. The Alter Table (ADB21TAB) panel opens, where you can enter the the column details (Figure 14-36).

```

ADB21TAB ----- DB0B Alter Table ----- 18:16
Command ==>

ALTER TABLE
Table schema . . ADMR3      >
Table name . . . BK_TO_CSTMR_STMT  >

ADD
Column name . . BK_TO_CSTMR_STMT_0 > (? to look up)
Column type . . XML                (Built-in only)
Data length . .                    (Built-in only)
Inline length .                    (0-32680 BLOB or CLOB, 0-16340 DBCLOB)
Precision . . .                    (used only w/FLOAT and DECIMAL)
Scale . . . . .                    (used only w/DECIMAL and TIMESTAMP)
Type schema . .                    > (User-defined only)
Type name . . .                    > (User-defined only)
WITH TIME ZONE .                    (Yes/No - for TIMESTAMP only)

Allow nulls . . YES (Yes or blank-nullable, No-NOT NULL)
FOR ? DATA . . . (B-Bit, S-SBCS, M-Mixed, blank-N/A)
WITH DEFAULT . . N (Yes, No, L (SECLABEL) or enter value below)
Default value . . . . . >
GENERATED . . . (A-ALWAYS, D-DFLT, I-ALWAYS AS IDENT, J-DFLT AS IDENT,

```

Figure 14-36 XML column details

[illegible]

This action generates the DDL to add the column to the table (Example 14-2).

```
ALTER TABLE ADMR3.BK_TO_CSTMRT_STMT ADD BK_TO_CSTMRT_STMT_OLD
XML(XMLSCHEMA ID SYSXSR.CAMT 053 001 02)
```

Example 14-3 XML Modifier in Create Table Statement

Chapter 14. Physical design 447

14.3.4 Creating an index on an XML column

As part of adding XML to your database schema, you might want to add indexes to your XML columns for performance reasons or uniqueness reasons. DB2 Administration Tool V10 can be used to build these indexes through the CREX Create Index command.

Open the table to which you want to add the index (Figure 14-38).

```
DB2 Admin ----- DBOB Tables, Views, and Aliases ----- Row 1 from 1
Command ==>                                         Scroll ==> PAGE

Commands: GRANT MIG ALL
Line commands:
C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
V - Views    T - Tables P - Plans  Y - Synonyms SEL - Select prototyping
? - Show all line commands

Sel   Name                Schema  T DB Name  TS Name   Cols      Rows Chks C
      *                  *      * *      *
-----
CREX  BK_TO_CSTMR_STMT    ADMR3   T ADMR3ADB TSAUDIT1   4         -1    0
***** END OF DB2 DATA *****
```

Figure 14-38 Creating an Index on an XML column

At the prompt, enter the name of the index you want to create, in this case, BK_TO_CSTMR_STMT_I (Figure 14-39).

```
ADB26CX n ----- DBOB Create Index ----- 16:39
Command ==>

CREATE INDEX

Schema . . . . ADMR3    >      (default is ADMR3)
Name   . . . . BK_TO_CSTMR_STMT_I > (? to look up)

ON
Table Schema . ADMR3    >      (default is ADMR3)
Table name   . . BK_TO_CSTMR_STMT > (? to look up)

Partitions . . 0                (0 for nonpartitioned INDEX)

Like:
Index Schema .          >      (required for Like usage)
Index name   . .        >      (? to look up)
```

Figure 14-39 Naming an index

Press Enter. The Create Index (ADB21XAR) panel opens, where you add the index features (Figure 14-40).

```

ADB21XAR ----- DB0B Create Index ----- Row 1 from 4
Command ==> Scroll ==> PAGE

Commands: CONTINUE ORIGINAL EXPRESSION XMLPATTERN
Line commands: nnn A|D - Sequence & order R - Remove the column I - Include
A - Ascending D - Descending RA - Random U - Update expression/XML pattern
B - Business Time without overlaps

CREATE INDEX ADMR3 . BK_TO_CSTMR_STMT_I >
ON ADMR3.BK_TO_CSTMR_STMT
Unique . . . . . YES Where Not Null . . . YES Cluster . . . . . NO
Buffer Pool . . . . BP8K0 Close Rule . . . . . NO Copy Allowed . . YES
Piece Size . . . . . 2G Define . . . . . YES Defer . . . . . NO
Partitioned . . . . . Padded . . . . . NO Compress . . . . . YES

Select Column Name Col Type Length Scale N ColSeq Ord
* * * * *
----->-----
MSG_ID VARCHAR 35 0 N
MSG_CRE_DT_TM TIMESTMP 10 6 N
BK_TO_CSTMR_STMT_O XML 14 0 Y
DB2_GENERATED_DOCI BIGINT 8 0 Y
***** END OF DB2 DATA *****

```

Figure 14-40 Create Index panel

Select the XML Column that you want to index and enter 1A (first column Ascending) on the of the XML column line (Figure 14-41).

```
ADB21XAR ----- DBOB Create Index ----- Row 1 from 4
Command ==> Scroll ==> PAGE

Commands: CONTINUE ORIGINAL EXPRESSION XMLPATTERN
Line commands: nnn A|D - Sequence & order R - Remove the column I - Include
A - Ascending D - Descending RA - Random U - Update expression/XML pattern
B - Business Time without overlaps

CREATE INDEX ADMR3 . BK_TO_CSTMRT_STMT_I >
ON ADMR3.BK_TO_CSTMRT_STMT
Unique . . . . . YES Where Not Null . . . YES Cluster . . . . . NO
Buffer Pool . . . . BP8K0 Close Rule . . . . NO Copy Allowed . . YES
Piece Size . . . . . 2G Define . . . . . YES Defer . . . . . NO
Partitioned . . . . . Padded . . . . . NO Compress . . . . YES

Select Column Name      Col Type      Length  Scale N ColSeq Ord
      *                  *              *      * *      * *
----->-----
*      BK_TO_CSTMRT_STMT_O XML              14      0 Y      1 A
      MSG_ID              VARCHAR              35      0 N
      MSG_CRE_DT_TM        TIMESTMP              10      6 N
      DB2_GENERATED_DOCI    BIGINT              8      0 Y
***** END OF DB2 DATA *****
```

Figure 14-41 Selecting index column

You have now created an index for the XML column, but if you want to use an XMLPATTERN, issue the XMLPATTERN command. The Create Index XML Pattern panel opens, where you enter the pattern that you want to use for the index (Figure 14-42).

```
ADB21XAX ----- DBOB Create Index XML Pattern ----- Columns 00001 00072
Command ==> Scroll ==> CSR

GENERATE KEYS USING XMLPATTERN <XML pattern-expression>
AS SQL <data-type>

Choose the data-type and length value for that data-type.
Data type . .      1. VARCHAR (1 to 1000)      3. DATE
                  2. DECFLOAT (blank or 34)    4. TIMESTAMP (blank or 12)
Data length .

XML pattern-expression (without quotes):
***** ***** Top of Data*****
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
000001
***** ***** Bottom of Data*****
```

Figure 14-42 Entering XMLPATTERN

Enter the pattern that you want to use (in this case, /Document/Bk_to_Cstmr_Stmt_0/GrpHdr/MsgId) and select the type of the column data as VARCHAR(35) (Figure 14-43).

```
ADB21XAX ----- DB0B Create Index XML Pattern ----- Columns 00001 00072
Command ==> Scroll ==> CSR

GENERATE KEYS USING XMLPATTERN <XML pattern-expression>
AS SQL <data-type>

Choose the data-type and length value for that data-type.
Data type . . 1      1. VARCHAR    (1 to 1000)      3. DATE
                2. DECFLOAT    (blank or 34)      4. TIMESTAMP (blank or 12)
Data length . 35

XML pattern-expression (without quotes):
***** ***** Top of Data *****
000001 /Document/Bk_to_Cstmr_Stmt_0/GrpHdr/MsgId
***** ***** Bottom of Data *****
```

Figure 14-43 XMLPATTERN within an index

When you return to the previous panel, you see that the XMLPATTERN line has been added (Figure 14-44).

```
ADB21XAR ----- DB0B Create Index ----- Row 1 from 5
Command ==> Scroll ==> PAGE

Commands: CONTINUE ORIGINAL
Line commands: nnn A|D - Sequence & order R - Remove the column I - Include
A - Ascending D - Descending RA - Random U - Update expression/XML pattern
B - Business Time without overlaps

CREATE INDEX ADMR3 . BK_TO_CSTMRT_STMT_I >
ON ADMR3.BK_TO_CSTMRT_STMT
Unique . . . . . YES Where Not Null . . . YES Cluster . . . . . NO
Buffer Pool . . . . BP8K0 Close Rule . . . . . NO Copy Allowed . . YES
Piece Size . . . . . 2G Define . . . . . YES Defer . . . . . NO
Partitioned . . . . . Padded . . . . . NO Compress . . . . . YES

Select Column Name Col Type Length Scale N ColSeq Ord
* * * * *
----->-----
* <XMLPATTERN> 1
BK_TO_CSTMRT_STMT_O XML 14 0 Y 1 A
MSG_ID VARCHAR 35 0 N
MSG_CRE_DT_TM TIMESTMP 10 6 N
DB2_GENERATED_DOCI BIGINT 8 0 Y
***** END OF DB2 DATA *****
```

Figure 14-44 Generating the XMLPATTERN index

You have now defined an index on an XML column using XMLPATTERN strings. Press Enter. The subsequent panels are the same as for a non-XML index creation, but the DDL that is executed is different (Example 14-4).

Example 14-4 XMLPATTERN Index DDL

```
CREATE UNIQUE WHERE NOT NULL INDEX
  "ADMR3"."BK_TO_CSTMR_STMT_I"
ON
  "ADMR3"."BK_TO_CSTMR_STMT"
  ( "BK_TO_CSTMR_STMT_OLD" )
  GENERATE KEYS USING XMLPATTERN '/Document/Bk_to_Cstmr_Stmt_0/GrpHdr/
MsgId' AS SQL VARCHAR(35)
  USING STOGROUP SYSDEFLT
  FREEPAGE 0
  PCTFREE 5
  DEFINE YES
  NOT PADDED
  COMPRESS YES
  BUFFERPOOL BP8K0
  CLOSE NO
  DEFER NO
  COPY YES
  PIECESIZE 2G
```

14.4 Adding an active log data set

DB2 10 for z/OS has the ability to add an active log data set without stopping DB2. The SET LOG command has two new parameters: NEWLOG and COPY.

To add a new active log using the DB2 Administration Tool, enter a Z on the Command line of the DB2 Administration Menu to access the System Administration menu. Enter an LZ - Set log checkpoint frequency command on the Command line (Figure 14-45).

```

DB2 Admin ----- DB0B System Administration ----- 19:30
Option ==> lz

DB2 System: DB0B
DB2 SQL ID: ADMR1
More: +

DB2 activity related functions:
  2D - Display threads                2U - Display/terminate utilities
  2T - Display/manage traces          2R - Display/update resource limits
  2S - Stop DB2                      2G - Display group
  2B - Display/manage batch checkpoint 2Z - Manage system parameters

Buffer pool functions:
  BD - Display buffer pools          BA - Alter buffer pools
  BH - Display buffer pool hit ratios

DB2 log functions:
  LD - Display archive log parameters LS - Set archive log parameters
  LA - Archive current log           LI - Display log information
  LZ - Set log checkpoint frequency

DDF functions:
  DU - Display/update CDB
  DC - Display/cancel distributed thds DL - Display active locations
  DT - Start DDF                      DS - Stop DDF

Stored procedures and functions options:

```

Figure 14-45 System Administration panel

The Change DB2 System Checkpoint Frequency panel opens. Two new parameters, NEWLOG and COPY, are at the bottom of the panel. (Figure 14-46).

```

DB2 Admin ----- DB0B Change DB2 System Checkpoint Frequency ----- 19:31
Command ==>

-SET LOG
Mode . . . . . b (S-SINGLE, B-BOTH)
LOGLOAD(
Checkpoint frequency . . (1000-16000000 when Mode=S,
) 0 or 1000-99999999 when Mode=B)

CHKTIME(
Checkpoint frequency . . (1-60 when Mode=S,
) 0-1439 when Mode=B)

SUSPEND . . . . . (Yes/No)
RESUME . . . . . (Yes/No)
NEWLOG . . . . .
COPY . . . . . (1/2)

```

Figure 14-46 Change DB2 System Checkpoint Frequency panel

Note: The active log data set must be pre-allocated before using the -SET LOG command with the NEWLOG and COPY options. The recommendation is to use DSNJLOGF to accomplish this task.

NEWLOG is used to add a new active log. COPY specifies the number of the new log. Once specified, a -SET LOG command is generated (Figure 14-47).

```
DB2 Admin ----- DBOB Statement Execution Prompt ----- 20:05
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'TEAM76.ALTER.WSL'
Work statement list name ==> TSALT      Action ==> A (Append or Replace)
                                           More:      +

Statement that is about to be executed (first 28 lines):
-SET LOG BOTH NEWLOG(DBOBL.LOGCOPY3.DS01) COPY(2)
```

Figure 14-47 SET LOG NEWLOG COPY

14.5 ADD PARTITION

DB2 10 for z/OS has the ability to add a partition to a table residing in a partitioned by growth table space by using the ALTER TABLE ... ADD PARTITION *n* syntax, where *n* represents the number of physical partitions to add.

To add a partition in DB2 Administration Tool V10, enter the ALT line command next to the name of the table residing in the partition-by-growth table space. Click the ADDPART keyword at the top of the panel or enter ADDPART on the Option line. Even though you are using ALT, DB2 Administration Tool generates an ALTER statement in the work statement list member (Figure 14-48).

```
DB2 Admin ----- Edit Statement ----- Columns 00001 00072
Command ==>                               Scroll ==> CSR

***** ***** Top of Data *****
000001 -- Created by DBA104 on 2010/11/15 at 17:21
000002 -- Generated by apply exec by DBA104 on 2010/11/15 at 17:21
000003 --#ADMIN PROCESS CREATE
000004  ALTER TABLE TEAM83.TD83TB11_PARTTB
000005      ADD PARTITION ;
000006 -- End of Apply statements
***** ***** Bottom of Data *****
```

Figure 14-48 ALTER TABLE ... ADD PART syntax

The number of partitions added cannot exceed the value specified in MAXPARTITIONS of the table space. If it does, an error message displays (Figure 14-49).

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
```

```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS  
EDIT DBA104.DBA104.ADBWORK.ADDPART2 Columns 00001 00072  
Command ==> Scroll ==> CSR  
  
000016 ALTER TABLE TEAM83.TD83TB11_PARTTB  
000017 ADD PARTITION ;  
000018 sqlerror on ALTER command, EXECUTE function  
000019 DSNT408I SQLCODE = -4701, ERROR: THE NUMBER OF PARTITIONS, OR THECOM  
000020 OF THE NUMBER OF TABLE SPACE PARTITIONS AND THECORRESPONDING  
000021 OF THE PARTITIONING LIMIT KEY EXCEEDS THE SYSTEM LIMIT,  
000022 OR OPERATION IS NOT ALLOWED ON SYSTEM DATABASES; OR THECOMBI  
000023 THE NUMBER OF TABLE SPACE PARTITIONS EXCEEDS THEMAXPARTITION  
000024 PARTITION BY GROWTH TABLE SPACES  
000025 DSNT418I SQLSTATE = 54054 SQLSTATE RETURN CODE  
000026 DSNT415I SQLERRP = DSNXIPGA SQL PROCEDURE DETECTING ERROR  
000027 DSNT416I SQLERRD = 31 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION  
000028 DSNT416I SQLERRD = X'0000001F' X'00000000' X'00000000' X'FFFFFFF  
000029 ADB5185E The run is ending due to errors.  
000030 ADBT010W - Rollback performed due to previous error  
000031 Additional diagnostics  
000032 NNNINNN  
000033 commit_all_req=N  
000034 asw_req =N  
000035 asw_app =N
```

Figure 14-49 Error message when exceeding the number of MAXPARTITIONS

Note: ALT is a valid line command for tables and table spaces.

Issuing the ALT command next to a table enables you to physically add a new partition to the partition by growth object, as long as the number of MAXPARTITIONS is not exceeded. This change can be accomplished by an ALTER command.

Issuing the ALT command next to a table space enables you to increase the number of MAXPARTITIONS. This change requires an unload of the data, followed by dropping and recreating the object and reloading the data.

14.6 TIMESTAMP with TIME ZONE

TIMESTAMP with TIME_ZONE is a new SQL data type in DB2 10 for z/OS. TIME_ZONE represents the difference of offset between local time and the Coordinated Universal Time (UTC), formerly known as Greenwich Mean Time (GMT), in hours and minutes and can be appended to the end of the time stamp when using the TIMESTZ data type.

Figure 14-50 shows a **TIMESTAMP** field. The format of the data is **YYYY-MM-DD-HR.MN.SEC.MICROSECOND**.

```

***** Top of Data *****
A1                               B1
-----
2010-10-01-09.00.00.000000 A
***** Bottom of Data *****

```

Figure 14-50 **TIMESTAMP: YYYY-MM-DD-HH.MN.SS.MICROSEC**

Figure 14-51 shows the new DB2 10 for z/OS **TIMESTAMP WITH TIME ZONE SQL** data type. The format is **YYYY-MM-DD-HH.MN.SS.MICROSEC-timezone HR:timezone MN**.

```

***** Top of Data *****
B1 A1
-----
T 2010-11-17-15.53.03.873783-06:00
p 2010-11-17-18.06.16.868853-06:00
p 2010-11-17-18.06.46.294882-06:00
P 2010-11-17-18.09.26.367420-06:00
D 2010-11-17-18.10.11.794601-06:00
D 2010-11-17-18.11.14.781026-06:00
a 2010-11-17-18.18.46.343714-06:00
a 2010-11-17-18.27.24.429335-06:00
a 2010-11-17-18.27.41.114945-06:00
A 2010-11-17-18.27.57.263011-06:00
A 2010-10-01-10.10.10.000000-06:00
A 2010-10-01-10.10.10.000000-06:00

```

Figure 14-51 **TIMESTAMP WITH TIME ZONE**

DB2 Administration Tool support for the **TIMESTAMP WITH TIME ZONE** feature can be found in the following locations:

- ▶ **AL Table - ADD Column**
- ▶ **ALT Table - Insert a column** (Use the U line command next to the new column name.)
- ▶ **CREATE Table**
- ▶ **CREATE Function**
- ▶ **CREATE Procedure**

To use the AL command to add a column to an existing table, enter AL next to the table name. Select the ADD column option on the Alter Table panel (Figure 14-52).

DB2 Admin ----- DSNT Alter Table -----
13:49
Command ==>

Table schema . . : DNET305 >
Table name . . . : TIMEZ3 >

AUDIT NONE (None, Changes, or All)
DATA CAPTURE NONE (None/Changes)
VALIDPROC NULL (NULL/Program name)
RESTRICT ON DROP . . . NO (Yes/No)
VOLATILE NO (Yes/No)
APPEND NO (Yes/No)

ALTER TABLE with any of the above changes OR select one of the options below

s **ADD column**
ADD PRIMARY KEY
DROP PRIMARY KEY
ADD FOREIGN KEY
DROP FOREIGN KEY
ADD CHECK constraint
DROP CHECK constraint
ADD UNIQUE constraint

ADD MATERIALIZED QUERY
DROP MATERIALIZED QUERY
REFRESH MATERIALIZED TABLE
ADD PARTITIONING KEY
ADD PARTITION
ADD CLONE
DROP CLONE
ADD VERSIONING

More: +

Figure 14-52 AL ADD column

Enter the name of the column, the data type, the length, and specify YES next to the WITH TIME ZONE key word under the ADD heading (Figure 14-53).

Note: If you specify the Column Type as being TIMESTAMP, the panel changes it to TIMESTZ, but the ALTER statement syntax reads TIMESTAMP length WITH TIME ZONE.

DB2 Admin ----- DSNT Alter Table -----		14:22
Command ==>		
		More: +
ALTER TABLE		
Table schema . .	DNET305	>
Table name . . .	TIMEZ3	>
ADD		
Column name . .	SAMPLETZ	> (? to look up)
Column type . .	TIMESTZ	(Built-in only)
Data length . .		(Built-in only)
Inline length .		(0-32680 BLOB or CLOB, 0-16340 DBCLOB)
Precision . . .		(used only w/FLOAT and DECIMAL)
Scale		(used only w/DECIMAL and TIMESTAMP)
Type schema . .		> (User-defined only)
Type name . . .		> (User-defined only)
WITH TIME ZONE .	YES	(Yes/No - for TIMESTAMP only)
Allow nulls . .		(Yes or blank-nullable, No-NOT NULL)
FOR ? DATA . .		(B-Bit, S-SBCS, M-Mixed, blank-N/A)
WITH DEFAULT . .		(Yes, No, L (SECLABEL) or enter value below)
Default value .		>
GENERATED . . .		(A-ALWAYS, D-DFLT, I-ALWAYS AS IDENT, J-DFLT AS IDENT,

Figure 14-53 Enter the Column name, Column type, Scale, and WITH TIME ZONE YES

The ALTER statement shown in Figure 14-54 is generated. The default scale is 6.

```
DB2 Admin ----- DSNT Statement Execution Prompt -----
14:02
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
    1 - Execute the statement
    2 - Edit the statement
    3 - Create a batch job with the statement
    4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'DBA104.ALTER.WSL'
Work statement list name ==> ALTERDB    Action ==> A (Append or Replace)
                                           More:      +

Statement that is about to be executed (first 28 lines):
ALTER TABLE "DNET305"."TIMEZ3"
ADD "SAMPLETZ" TIMESTAMP (6) WITH TIME ZONE
```

Figure 14-54 ALTER statement

If you add a column with the **TIMESTAMP WITH TIME ZONE** data type to a table using the **ALT** command, you only need to specify the name of the column, the column type **TIMESTZ**, the length of the column, the scale, and the nullability options. DB2 Administration Tool makes the change using the new SQL syntax (Figure 14-55).

```
DB2 Admin ----- DSNT Statement Execution Prompt ----- 14:29
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now (Add an A for all stmts. For example 1A - Execute all stmts):
    1 - Execute the statement
    2 - Edit the statement
    3 - Create a batch job with the statement
    4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'DBA104.ALTER.WSL'
Work statement list name ==> ALTERDB    Action ==> A (Append or Replace)
                                           More:      +

Statement that is about to be executed (first 28 lines, more stmts pending):
ALTER TABLE DNET305.TIMEZ3
ADD
    SAMPLETZ2                TIMESTAMP (6) WITH TIME ZONE
    WITH DEFAULT NULL
```

Figure 14-55 SQL statement generated to add a **TIMESTAMP WITH TIME ZONE** column using **ALT**

You can determine that the time zone has been appended to a `TIMESTAMP` column if the data type is `TIMESTZ` (Figure 14-56).

DB2 Admin -- DSNT Columns in Table DNET305.TIMEZ3									
Command ==>									
----- Row 1 to 3 of 3									
Scroll ==> CSR									
Line commands:									
T - Tables X - Indexes A - Auth GR - Grant H - Homonyms I - Interpret									
UR - Update runstats LAB - Label COM - Comment DI - Distribution stats									
? - Show all line commands									
Select	Column Name	Col	No	Col	Type	Length	Scale	Null	Def FP
*		*	*			*		*	* * *

	B1	1			VARCHAR	1		0 Y	Y N -1
	A1	2			TIMESTZ	12		6 Y	Y N -1
	SAMPLETZ	3			TIMESTZ	12		6 Y	Y N -1
***** END OF DB2 DATA *****									

Figure 14-56 `TIMESTAMP WITH TIME ZONE` Column Type `TIMESTZ`

14.7 CONCENTRATE STATEMENTS WITH LITERAL

`CONCENTRATE STATEMENTS WITH LITERAL` is a new clause of the `PREPARE` statement. If it is specified, more dynamic SQL statements stored in the dynamic SQL Cache can be shared. Previously, if a dynamic SQL statement contained literals, the odds were that the statement would not be reused as the literals changed. With DB2 10 for z/OS and `CONCENTRATE STATEMENTS WITH LITERAL` turned on, if the only difference in the statement in the cache and a new SQL statement are the literals, then the literals are replaced with an `&` and you can avoid having to do a `PREPARE`.

You can set this special register at customization time. `ADB2CUST` has a new parameter that can be set to Yes for the subsystem. The default is Yes.

14.8 USE CURRENTLY COMMITTED

In prior versions of DB2, applications doing reads acquired locks. As a result, committed data from update transactions was not available for use until all the locks were released. With DB2 10 for z/OS, there is a new option that can be used on the `BIND` and `PREPARE` statements that enable the committed data to be returned without having to wait for the `LOCKS` to be freed. This also works for `INSERTS` and `DELETES`.

There are two options:

- ▶ `USE CURRENTLY COMMITTED`
- ▶ `WAIT FOR OUTCOME`

This value can be set in the `ADB2CUST :currcom` parameter.

It is also a parameter for the BIND statement (Figure 14-57).

```

DB2 Admin ----- DSNT Bind Application Plan ----- 17:15
Command ==>

                                     More:  - +
DISABLE . . . . . (use ? to get current values from the catalog)
En/disable names . . (use ? to get current values from the catalog)

DEGREE . . . . . 1 (1 or ANY) (Parallelism)
SQLRULES . . . . . D (DB2 or STD)
DISCONNECT . . . . . E (Explicit, Automatic, or Conditional)
DYNAMICRULES . . . . (Run or Bind)
KEEPDYNAMIC . . . . NO (Yes/No)
REOPT(VAR) . . . . NONE (N - None, Y - Always, 1 - Once, or A-Auto)

OPTHINT . . . . . >
PATH . . . . . >
ENCODING . . . . . 37 (ASCII, EBCDIC, UNICODE or ccsid)
IMMEDWRITE . . . . . NO (Yes, No or PH1)
ROUNDING . . . . . HALFEVEN (Ceiling, Down, Floor, HalfDown,)
                                     (HalfEven, HalfUp or Up)
CONCURRENTACCESSRES (U - Usecurrentlycommitted or)
                                     (W - Waitforoutcome)

```

Figure 14-57 CONCURRENTACCESSRES

Note: The parameter enables the data to be returned, but there is no guarantee that DB2 will allow it. There are some situations where DB2 reverts to unconditional locking.

14.9 ALTER BUFFERPOOL PAGESTEAL

PAGESTEAL is a new option for ALTER BUFFERPOOL. When set to NONE, no pages can be stolen. All pages remain resident in the buffer pool.

This option is available on the DB2 Administration Tool V10 Alter Buffer Pool panel, which is a subpanel to the DB2 System Administration panel.

To navigate to the DB2 System Administration panel, start at the DB2 Administration Menu and enter a Z on the Option line (Figure 14-58).

```
DB2 Admin ----- DB2 Administration Menu 10.1.0 ----- 17:31
Option ==> z

    1 - DB2 system catalog                      DB2 System: DSNT
    2 - Execute SQL statements                  DB2 SQL ID: DBA104
    3 - DB2 performance queries                 Userid   : DBA104
    4 - Change current SQL ID                   DB2 Schema: DBA104
    5 - Utility generation using LISTDEFS and  DB2 Rel   : 1015
        TEMPLATES
    P - Change DB2 Admin parameters
    DD - Distributed DB2 systems
    E - Explain
    Z - DB2 system administration
    SM - Space management functions
    W - Manage work statement lists
    X - Exit DB2 Admin
    CC - DB2 catalog copy version maintenance
    CM - Change management

                                                    More:      +

Interface to other DB2 products and offerings:
    I  DB2I
    C  DB2 Object Comparison Tool
```

Figure 14-58 DB2 Administration Tool Menu panel

To alter a buffer pool, enter BA on the Option line of the System Administration pane (Figure 14-59).

```
DB2 Admin ----- DSNT System Administration ----- 17:35
Option ==> ba

DB2 System: DSNT
DB2 SQL ID: DBA104
More:      +

DB2 activity related functions:
  2D - Display threads                2U - Display/terminate utilities
  2T - Display/manage traces          2R - Display/update resource limits
  2S - Stop DB2                      2G - Display group
  2B - Display/manage batch checkpoint 2Z - Manage system parameters

Buffer pool functions:
  BD - Display buffer pools          BA - Alter buffer pools
  BH - Display buffer pool hit ratios

DB2 log functions:
  LD - Display archive log parameters  LS - Set archive log parameters
  LA - Archive current log            LI - Display log information
  LZ - Set log checkpoint frequency

DDF functions:
  DU - Display/update CDB
  DC - Display/cancel distributed thds  DL - Display active locations
  DT - Start DDF                      DS - Stop DDF

Stored procedures and functions options:
```

Figure 14-59 Alter buffer pools

To alter the page steal option, enter AL next to the name of the buffer pool to be changed (Figure 14-60).

DB2 Admin ----- DSNT Alter Buffer Pools ----- Row 1 to 14 of 80										
Command ==> Scroll ==> CSR										
Line commands:										
AL - Alter buffer pool DIS - Display buffer pool										
BP	VP	VP	PG	Int1	Int2	VP X	Auto			
Se1 Name	Size		Steal	SEQT	PSEQT	FIX	DWQT	VDWQT	VDWQT	PSEQT Size
*	*		*	*	*	*	*	*	*	*

	BP0	3000	LRU	80	50	NO	30	5	0	0 NO
	BP1	1000	LRU	80	50	NO	30	5	0	0 NO
a1	BP2	5000	LRU	80	50	NO	30	5	0	0 NO
	BP3	750	LRU	80	50	NO	30	5	0	0 NO
	BP4	1000	LRU	80	50	NO	30	5	0	0 NO
	BP5	0	LRU	80	50	NO	30	5	0	0 NO
	BP6	0	LRU	80	50	NO	30	5	0	0 NO
	BP7	0	LRU	80	50	NO	30	5	0	0 NO
	BP8	0	LRU	80	50	NO	30	5	0	0 NO
	BP9	0	LRU	80	50	NO	30	5	0	0 NO
	BP10	0	LRU	80	50	NO	30	5	0	0 NO
	BP11	0	LRU	80	50	NO	30	5	0	0 NO
	BP12	0	LRU	80	50	NO	30	5	0	0 NO
	BP13	0	LRU	80	50	NO	30	5	0	0 NO

Figure 14-60 Alter Buffer Pools

The options are LRU, FIFO, and NONE (Figure 14-61).

DB2 Admin ----- DSNT Alter Buffer Pool ----- 17:39										
Command ==>										
-ALTER BUFFERPOOL(
Buffer pool name BP2 (BP0-49, BP8K0-9, BP16K0-9, BP32K0-9)										
VPSIZE 5000 (0-250MB for 4K, 0-31250KB for 32K)										
PGSTEAL LRU (L-LRU, F-FIFO or N-NONE)										
PGFIX NO (Yes/No)										
VPSEQT 80 (0-100)										
VPPSEQT 50 (0-100)										
DWQT 30 (0-90)										
VDWQT										
Vertical DWQT, integer1 . 5 (0-90)										
Vertical DWQT, integer2 . 0 (0-9999)										
VPXPSEQT 0 (0-100)										
AUTOSIZE NO (Yes/No)										

Figure 14-61 Alter Buffer Pool



Online schema evolution support

This chapter discusses the functions provided by DB2 Administration Tool for supporting the new online schema changes provided by DB2.

This chapter contains the following topics:

- ▶ Online schema evolution and DB2 pending changes
- ▶ DB2 Administration Tool support for DB2 pending changes

15.1 Online schema evolution and DB2 pending changes

DB2 10 for z/OS has made enhancements to the online schema evolution (ALTER) capabilities against table spaces. Changes may be specified to MEMBER CLUSTER, DSSIZE, SEGSIZE, and the page size options of a table space while keeping the application operational. As a consequence of this type of change, the object(s) are placed in the advisory reorg pending state (AREOR). This is not a restrictive state.

Note: It is the users responsibility to determine when to run a REORG to remove the AREOR state.

There are two types of ALTER changes in DB2 10 for z/OS:

- ▶ Immediate
- ▶ Pending

A change is said to be immediate when an ALTER is executed and the change is completed in its entirety. The objects impacted by the change are not placed in a pending state.

A DB2 pending change transpires when modifications are made to a table space that result in the generation of an ALTER statement that, when executed, puts the object in AREOR state. The change is not materialized or completed until a REORG SHRLEVEL CHANGE or REORG SHRLEVEL REFERENCE has run. Information about the pending change is inserted into a new DB2 table, SYSIBM.SYSPENDINGDDL.

DB2 has placed some restrictions on pending changes. All of the changes in one ALTER statement must be of the same type. In other words, all immediate or all pending. If they are not either one, a -20385 return code is displayed, which indicates that the changes are unable to be completed due to this DB2 restriction; there is also an accompanying reason code.

Here is an example of a valid pending change. A request has come in to alter the buffer pool and MAXPARTITIONS of a table space. BP0 is changed to BP8K0, so the page size changes as well. MAXPARTITIONS is changed from 0 to 20. In both cases, the ALTER places the object in the AREOR state. Thus, this is a valid combination of changes. If one of the changes had been something that could be executed immediately, though, such as changing the buffer pool to another buffer pool with equivalent page sizes, the combination would be flagged as invalid because you cannot combine different change types.

Note: Pending changes only occur under the right set of circumstances. Any deviation is not tolerated.

15.2 DB2 Administration Tool support for DB2 pending changes

The AL line command provides the means for specifying changes to a table space, which can be achieved by using an ALTER SQL statement. DB2 Administration Tool supports both immediate and pending changes. Pending changes follow the rules specified by DB2. If the user makes a modification that results in a mixture of change types, the result will be a -20385 return code.

A new primary command, PDC (DB2 pending definition changes), has been added to the System Catalog panel (Figure 15-1).

```

ADB21 min ----- DB0B System Catalog ----- 17:46
Option ==> pdc

Object options:
  AO - Authorization options
  G - Storage groups
  D - Databases
  S - Table spaces
  T - Tables, views, and aliases
  V - Views
  A - Aliases
  Y - Synonyms
  X - Indexes
  C - Columns
  N - Constraints
  DS - Database structures
  P - Plans
  L - Collections
  K - Packages
  M - DBRMs
  H - Schemas
  E - User defined data types
  F - Functions
  O - Stored procedures
  J - Triggers
  Q - Sequences
  DSP - DS with plans and packages
  More: +
  DB2 System: DB0B
  DB2 SQL ID: ADMR1

PDC - DB2 pending definition changes

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name ==> SS% > Grantor ==> >
Owner ==> > Grantee ==> >
In D/L/H ==> >
And/or other selection criteria (option xC shows you columns for option x)

```

Figure 15-1 System Catalog PDC command

PDC produces a listing of the entries in the SYSIBM.SYSPENDINGDDL table. Filters specified on the bottom of the System Catalog panel apply to the Pending Definition Changes panel too (Figure 15-2).

```

ADBPPDC n ----- DB0B DB2 Pending Definition Changes----- Row 1 to 2 of 2
Command ==> Scroll ==> PAGE

Commands: DIS UTIL DROP
Line commands:
  T - Tables D - Database X - Indexes S - Table spaces UTIL - Utilities
  DIS - Display object DROP - Drop changes SQ - Statement text I -Interpret
  ? - Show all line commands

Select Name          Qualifier T S Keyword          Value          Timestamp
      *              *      * *
-----
      SS12DPT        SSEMDBB  S 1 MAXPARTITIONS  20             2010-11-09-14
      SS12DPT        SSEMDBB  S 2 BUFFERPOOL    BP8K0          2010-11-09-14
***** END OF DB2 DATA *****

```

Figure 15-2 DB2 Pending Definition Changes panel

There are a number of actions that can be taken on this panel, such as DROP Changes. DROP can be issued as a line command or a primary command. If issued as a primary command, all of the ending changes displayed on the panel are dropped. If issued as a line command, the changes on that line are dropped.

When performing a DROP, the user is asked to confirm the action (Figure 15-3).

```

ADBPPDCC ----- DB0B Drop DB2 Pending Definition Changes con Row 1 to 2 of 2
Command ==>                                     Scroll ==> PAGE

DROP DB2 PENDING DEFINITION CHANGES

Here are all the DB2 pending definition changes to be dropped:

Confirm drop: yes (Yes/No)

Select Name                Qualifier T S Keyword          Value          Timestamp
      *                    *      * *
-----
      SS12DPT              SSEMDBB  S 1 MAXPARTITIONS  20             2010-11-09-14
      SS12DPT              SSEMDBB  S 2 BUFFERPOOL    BP8KO          2010-11-09-14
***** END OF DB2 DATA *****

```

Figure 15-3 Drop DB2 Pending Definition Changes

If confirmed, an ALTER statement is generated (Figure 15-4).

```

ADB2PSTM ----- DB0B Statement Execution Prompt ----- 17:57
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'TEAM76.ALTER.WSL'
Work statement list name ==> TSALT      Action ==> A (Append or Replace)
                                           More:      +

Statement that is about to be executed (first 28 lines):
ALTER TABLESPACE "SSEMDBB"."SS12DPT" DROP PENDING CHANGES

```

Figure 15-4 ALTER TABLESPACE with DROP PENDING CHANGES

Another command that may be of interest when working with a DB2 pending change is the I or Interpret line command. This command displays detail information about the specific line item on the panel (Figure 15-5).

```
ADBPPDCI ----- DBOB Interpretation of Object in SYSPENDINGDDL ----- 18:07
Option ==>

Details for object . . . : "SSEMMDBB"."SS12DPT"

Object schema . . . . . : ADMR1
Object ID (OBID) . . . . : 39
Table space name . . . . : SS12DPT
Table space ID (PSID) . . : 39
Database name . . . . . : SSEMMDBB
Database ID (DBID) . . . : 566
Created at . . . . . : 2010-11-09-14.41.33.465664
Object type . . . . . : S - Table space
Statement type . . . . . : A - Alter
Environment ID . . . . . : 25
Sequence number . . . . . : 1
Keyword . . . . . : MAXPARTITIONS
Value . . . . . : 20
Created in DB2 Version . : 0 - DB2 V10
Statement text (first 250 characters):
ALTER TABLESPACE SSEMMDBB.SS12DPT .. BUFFERPOOL BP8K0 ..
MAXPARTITION
```

Figure 15-5 Interpret line command

There is also an SQ line command that displays the SQL for the pending change (Figure 15-6).

```
ADB2DF in -- Browse Result of SQL Select ----- Row 1 to 2 of 2
Command ==> Scroll ==> PAGE

L STATEMENT_TEXT
*
- -----
ALTER TABLESPACE SSEMMDBB.SS12DPT .. BUFFERPOOL BP8K0 ..
MAXPARTITIO
ALTER TABLESPACE SSEMMDBB.SS12DPT .. BUFFERPOOL BP8K0 ..
MAXPARTITIO
***** END OF DB2 DATA *****
```

Figure 15-6 SQ line command

The completion or materialization of a pending change requires a REORG. To remove the AREOR pending state, you must run SHRLEVEL REFERENCE or SHRLEVEL CHANGE REORG.

The following sections contain some scenarios demonstrating how DB2 Administration Tool supports this feature of DB2 10 for z/OS.

15.2.1 Scenario 1: Changes that result in a pending state

In this scenario, we change the page size of the buffer pool (Figure 15-7) and specify a value for MAXPARTITIONS. Enter AL next to a table space. Figure 15-19 on page 479 shows the original definition stored in the DB2 Catalog. BP is set to BP16 and MAXPARTITIONS is set to 0.

```

DB2 Admin ----- DB0B Alter Table Space ----- Row 1 to 1 of 1
Command ==>                                         Scroll ==> PAGE

Line commands:
D - Display Database   I - Interpret

ALTER TABLESPACE : SSEMMDDB.SS12DPT      (Nonpartitioned      )

Buffer Pool . . . . BP16      Close Rule . . . YES Max Rows . . 255
Lock Size . . . . . ANY                      Lock Max . . SYSTEM
Max Partitions . . . 0      LOG . . . . . YES

      Primary      Secondary      Free Pct  Com E T S
S   Part Quantity  Quantity      Page Free  prs R M T VCAT      StogroupGBPCache
- - - - -
      0            12            12      0   5 NO  N Y I DB0BD      SYSDEFLT CHANGED
***** END OF DB2 DATA *****

```

Figure 15-7 Original definition of the table space

Change BP16 to BP8K0 (8k page BP) and set MAXPARTITIONS to 20 (Figure 15-8).

```

DB2 Admin ----- DBOB Alter Table Space ----- Row 1 to 1 of 1
Command ==>                                           Scroll ==> PAGE

Line commands:
  D - Display Database   I - Interpret

ALTER TABLESPACE : SSEMMDDB.SS12DPT          (Nonpartitioned          )

Buffer Pool . . . . BP8K0          Close Rule . . . YES Max Rows . . 255
Lock Size . . . . . ANY                      Lock Max . . SYSTEM
Max Partitions . . . 20          LOG . . . . . YES

          Primary      Secondary   Free Pct  Com E T S
S  Part Quantity      Quantity    Page Free prs R M T VCAT      Stogroup
GBPCache
- -----> -----

          0          12          12    0    5 NO  N Y I DBOBD      SYSDEFLT CHANGED
***** END OF DB2 DATA *****

```

Figure 15-8 Modified table space definition

Our changes are complete, so press Enter, and assuming that PROMPT is set to Yes, the Statement Execution Prompt panel opens with the ALTER statement listed at the bottom of the panel (Figure 15-9).

```

DB2 Admin ----- DBOB Statement Execution Prompt ----- 14:37
Option ==> 1

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'TEAM76.ALTER.WSL'
Work statement list name ==> TSALT      Action ==> A (Append or Replace)
                                           More:      +

Statement that is about to be executed (first 28 lines):
ALTER TABLESPACE SSEMMDDB.SS12DPT
  BUFFERPOOL BP8K0
  MAXPARTITIONS 20

```

Figure 15-9 Statement Execution Prompt panel

The ALTER statement can be executed by using option 1, stored in a batch job by using option 3, or put into a work statement list member by using option 4. For demonstration purposes, we entered a 1 on the Option line to execute the ALTER. The ALTER occurs, and the Alter Table Space panel is displayed with a +610 message under the Command line at the top of the panel. This indicates that the object is in a pending state. (Figure 15-10).

```
DB2 Admin ----- DB0B Alter Table Space ----- Row 1 to 1 of 1
Command ==>                                           Scroll ==> PAGE
SQL warn(+610)

Line commands:
  D - Display Database   I - Interpret

ALTER TABLESPACE : SSEMMDBB.SS12DPT          (Nonpartitioned          )

Buffer Pool . . . . BP16          Close Rule . . . YES Max Rows . . 255
Lock Size . . . . . ANY                      Lock Max . . SYSTEM
Max Partitions . . . 0                LOG . . . . . YES

      Primary      Secondary      Free Pct      Com E T S
S   Part Quantity   Quantity      Page Free      prs R M T VCAT      StogroupGBPCache
- - - - -
      0             12             12    0    5 NO  N Y I DB0BD      SYSDEFLT CHANGED
*****
***** END OF DB2 DATA *****
```

Figure 15-10 AL Table Space with +610 message

To confirm this message, press F3 to return to the Table Spaces panel and issued a DIS command next to the name of the table space (Figure 15-11).

DB2 Admin ----- DB0B Table Spaces ----- Row 1 to 12 of 16									
Command ==> Scroll ==> PAGE									
Commands: GRANT MIG DIS STA STO ALL									
Line commands:									
T - Tables D - Database A - Auth G - Storage group ICS - Image copy									
status									
DIS - Display table space STA - Start table space STO - Stop table space									
? - Show all line commands									
Select	Name	DB Name	Parts	Bpool	L	E	S	I	C Tables
*	*	*	*	*	*	*	*	*	*

	SS12001	SSEMMDBB	0	BP15	A	N	A	N	Y
	SS12002	SSEMMDBB	0	BP15	A	N	A	N	Y
dis	SS12DPT	SSEMMDBB	0	BP16	A	N	A	N	Y
	SS12EMP	SSEMMDBB	4	BP15	A	N	A	N	N
	SS12EPA	SSEMMDBB	0	BP15	A	N	A	N	Y
	SS12PJA	SSEMMDBB	0	BP15	A	N	A	N	Y
	SS12PRJ	SSEMMDBB	0	BP15	A	N	A	N	Y
	SS12SPL	SSEMMDBB	7	BP15	A	N	A	N	Y
	SS13001	SSEMMDBC	0	BP15	A	N	A	N	Y
	SS13002	SSEMMDBC	0	BP15	A	N	A	N	Y
	SS13DPT	SSEMMDBC	0	BP16	A	N	A	N	Y
	SS13EMP	SSEMMDBC	4	BP1	A	N	A	N	Y

Figure 15-11 DIS command

The table space is in the AREOR state (Figure 15-12).

```
DB2 Admin ----- DBOB Browse DB2 Command Output --- Line 00000000 Col 001080
Command ==>                                           Scroll ==> PAGE

-DIS DB(SSEMMDBB) SPACENAM(SS12DPT) LIMIT(*)

***** Top of Data *****
DSNT360I -DBOB *****
DSNT361I -DBOB * DISPLAY DATABASE SUMMARY
              * GLOBAL
DSNT360I -DBOB *****
DSNT362I -DBOB DATABASE = SSEMMDBB STATUS = RW
              DBD LENGTH = 36332
DSNT397I -DBOB
NAME      TYPE PART  STATUS          PHYERRLO PHYERRHI CATALOG  PIECE
-----
SS12DPT TS RW,AREOR
***** DISPLAY OF DATABASE SSEMMDBB ENDED *****
DSN9022I -DBOB DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***** Bottom of Data *****
```

Figure 15-12 Results of the DIS command

At this point, the physical changes have not been made to the DB2 Catalog. Press F3 to return to the Table Spaces panel. Enter the DDL line command to extract the DDL from the DB2 catalog (Figure 15-13).

DB2 Admin ----- DB0B Table Spaces ----- Row 1 to 12 of 16

Command ==> Scroll ==> PAGE

Commands: GRANT MIG DIS STA STO ALL

Line commands:

T - Tables D - Database A - Auth G - Storage group ICS - Image copy

status

DIS - Display table space STA - Start table space STO - Stop table space

? - Show all line commands

Select	Name	DB Name	Parts	Bpool	L	E	S	I	C	Tables	Act.	pages	Segsz	T	L
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
	SS12001	SSEMMDBB	0	BP15	A	N	A	N	Y	7		-1	4	Y	
	SS12002	SSEMMDBB	0	BP15	A	N	A	N	Y	4		-1	4	Y	
DDL	SS12DPT	SSEMMDBB	0	BP16	A	N	A	N	Y	1		-1	32	Y	
	SS12EMP	SSEMMDBB	4	BP15	A	N	A	N	N	1		-1	4	R	Y
	SS12EPA	SSEMMDBB	0	BP15	A	N	A	N	Y	1		-1	64	Y	
	SS12PJA	SSEMMDBB	0	BP15	A	N	A	N	Y	1		-1	32	Y	
	SS12PRJ	SSEMMDBB	0	BP15	A	N	A	N	Y	1		-1	32	Y	
	SS12SPL	SSEMMDBB	7	BP15	A	N	A	N	Y	1		-1	4	R	Y
	SS13001	SSEMMDBC	0	BP15	A	N	A	N	Y	7		-1	4	Y	
	SS13002	SSEMMDBC	0	BP15	A	N	A	N	Y	4		-1	4	Y	
	SS13DPT	SSEMMDBC	0	BP16	A	N	A	N	Y	1		-1	32	Y	
	SS13EMP	SSEMMDBC	4	BP1	A	N	A	N	Y	1		-1	4	R	Y

Figure 15-13 Table Spaces

Note that the extracted DDL definition reflects the original definition. The BP is still BP16 and MAXPARTITIONS is not listed (Figure 15-14).

[illegible]

Figure 15-14 DDL for Table Space

Scroll down using F8 to see the DB2 pending change (Figure 15-15).

[illegible]

Figure 15-15 DDL for Table Space (continued)

Press F3 two times to return to the DB2 Administration Tool System Catalog panel (Figure 15-16).

```
DB2 Admin ----- DB0B System Catalog ----- 14:54
Option ==>

Object options:
A0 - Authorization options
G - Storage groups
D - Databases
S - Table spaces
T - Tables, views, and aliases
V - Views
A - Aliases
Y - Synonyms
X - Indexes
C - Columns
N - Constraints
DS - Database structures
P - Plans
L - Collections
K - Packages
M - DBRMs
H - Schemas
E - User defined data types
F - Functions
O - Stored procedures
J - Triggers
Q - Sequences
DSP - DS with plans and packages

More: +
DB2 System: DB0B
DB2 SQL ID: ADMR1

PDC - DB2 pending definition changes

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name ==> > Grantor ==> >
Owner ==> > Grantee ==> >
In D/L/H ==> >
And/or other selection criteria (option xC shows you columns for option x)
```

Figure 15-16 System Catalog panel

Enter the PDC command on the Option line at the top of the panel and press enter to display the DB2 Pending Definition Changes panel (Figure 15-17).

```
DB2 Admin ----- DB0B DB2 Pending Definition Changes----- Row 2 from 7
Command ==> Scroll ==> PAGE

Commands: DIS UTIL DROP
Line commands:
T - Tables D - Database X - Indexes S - Table spaces UTIL - Utilities
DIS - Display object DROP - Drop changes SQ - Statement text I -Interpret
? - Show all line commands

Select Name Qualifier T S Keyword Value Timestamp
* * * *
-----
SS12DPT SSEMMDBB S 1 MAXPARTITIONS 20 2010-11-09-14
SS12DPT SSEMMDBB S 2 BUFFERPOOL BP8KO 2010-11-09-14
***** END OF DB2 DATA *****
```

Figure 15-17 DB2 Pending Definition Changes panel

The changes have not physically taken place yet, but they are saved in the SYSPENDINGDDL DB2 table. Only when a REORG SHRLEVEL CHANGE or SHRLEVEL REFERENCE has run will the change become final.

15.2.2 Scenario 2: Mixing change types

This scenario demonstrates what happens in DB2 Administration Tool if there is a “mixed” change defined. In this instance, the buffer pool is changed to another buffer pool with the same page size and a value for MAXPARTITIONS was specified.

To start, enter the AL line command next to the name of a table space (Figure 15-18).

DB2 Admin ----- DB0B Table Spaces ----- Row 11 to 16 of 16									
Command ==> Scroll ==> PAGE									
Commands: GRANT MIG DIS STA STO ALL									
Line commands:									
T - Tables D - Database A - Auth G - Storage group ICS - Image copy									
status									
DIS - Display table space STA - Start table space STO - Stop table space									
? - Show all line commands									
Select	Name	DB Name	Parts	Bpool	L	E	S	I	C Tables
*	*	*	*	*	*	*	*	*	*

AL	SS13DPT	SSEMMDBC	0	BP15	A	N	A	N	Y
	SS13EMP	SSEMMDBC	4	BP1	A	N	A	N	Y
	SS13EPA	SSEMMDBC	0	BP15	A	N	A	N	Y
	SS13PJA	SSEMMDBC	0	BP15	A	N	A	N	Y
	SS13PRJ	SSEMMDBC	0	BP15	A	N	A	N	Y
	SS13SPL	SSEMMDBC	7	BP15	A	N	A	N	Y
***** END OF DB2 DATA *****									

Figure 15-18 AL line command next to a table space

In the original table space definition, the buffer pool BP15 and MAXPARTITIONS is set to 0 (Figure 15-19).

```

DB2 Admin ----- DB0B Alter Table Space ----- Row 1 to 1 of 1
Command ==>                                           Scroll ==> PAGE

Line commands:
  D - Display Database  I - Interpret

ALTER TABLESPACE : SSEMDBC.SS13DPT      (Nonpartitioned      )

Buffer Pool . . . . BP15      Close Rule . . . YES Max Rows . . 255
Lock Size . . . . . ANY                      Lock Max . . SYSTEM
Max Partitions . . . 0      LOG . . . . . YES

      Primary      Secondary      Free Pct  Com E T S
S  Part Quantity    Quantity    Page Free prs R M T VCAT      Stogroup
GBPCCache
- -----> -----

      0          12          12    0    5 NO  N Y I DB0BD      SYSDEFLT CHANGED
***** END OF DB2 DATA *****

```

Figure 15-19 Original definition

Change the buffer pool from BP15 to BP16. Because both buffer pools are the same page size, this change should be able to be done using an ALTER without any repercussions to the state of the object. Next, change MAXPARTITIONS from 0 to 20 (Figure 15-20).

```

DB2 Admin ----- DB0B Alter Table Space ----- Row 1 to 1 of 1
Command ==>                                           Scroll ==> PAGE

Line commands:
  D - Display Database  I - Interpret

ALTER TABLESPACE : SSEMDBC.SS13DPT      (Nonpartitioned      )

Buffer Pool . . . . BP16      Close Rule . . . YES Max Rows . . 255
Lock Size . . . . . ANY                      Lock Max . . SYSTEM
Max Partitions . . . 20      LOG . . . . . YES

      Primary      Secondary      Free Pct  Com E T S
S  Part Quantity    Quantity    Page Free prs R M T VCAT      StogroupGBPCCache
- -----> -----

      0          12          12    0    5 NO  N Y I DB0BD      SYSDEFLT CHANGED
***** END OF DB2 DATA *****

```

Figure 15-20 Updated fields

An ALTER statement is generated (Figure 15-21).

```
DB2 Admin ----- DB0B Statement Execution Prompt -----
17:39
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'TEAM76.ALTER.WSL'
Work statement list name ==> TSALT      Action ==> A (Append or Replace)
                                           More:      +

Statement that is about to be executed (first 28 lines):
ALTER TABLESPACE SSEMDBC.SS13DPT
  BUFFERPOOL BP16
  MAXPARTITIONS 20
```

Figure 15-21 ALTER statement generated

Enter a 1 (Execute the statement) on the Option line to execute the ALTER DDL at the bottom of the panel. A message with the negative return code -20385 is displayed because the changes requested were mixed. You cannot have an immediate type of change and a pending change in the same change request (Figure 15-22).

```
DB2 Admin ----- DB2 Error Display 1 ----- 17:41
Command ==>
Rollback done
      SQLCODE : -20385                                DSNTIAR CODE : 0

DSNT408I SQLCODE = -20385, ERROR:  THE STATEMENT CANNOT BE PROCESSED BECAUSE
      THERE ARE PENDING DEFINITION CHANGES FOR OBJECT SSEMDBC.SS13DPT OF
      TYPE TABLESPACE (REASON 1)
DSNT418I SQLSTATE  = 57007 SQLSTATE RETURN CODE
DSNT415I SQLERRP   = DSNXIATS SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD   = 470 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD   = X'000001D6' X'00000000' X'00000000' X'FFFFFFFF'
      X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION
```

Figure 15-22 Negative return code

Press F3 to remove the message. At this point, you have a couple of options:

- ▶ Edit the statement by keying by specifying a 2 on the Option line. Remove the MAXPARTITIONS parameter from the DDL. Now you only have an immediate change in the DDL. Press F3 to return to the Statement Execution Prompt panel. Execute the immediate ALTER by entering a 1 on the Option line. After the change has been completed, re-specify the MAXPARTITIONS parameter and execute that change separately.

- Save the original ALTER statement to a work statement list library member. Issue the WSL command to display the library and edit the statement by breaking it up into two ALTER statements, one for the BP and another for the MAXPARTITIONS. Run the WSL member either online or in batch. One change completes and the other is flagged as a pending change.

Upon completion of the ALTER MAXPARTITIONS statement, a +610 warning message is displayed, indicating that a change has put an object into AREOR pending state (Figure 15-23).

```
DB2 Admin ----- DB0B Alter Table Space ----- Row 1 to 1 of 1
Command ==>                                           Scroll ==> PAGE
SQL warn(+610)

Line commands:
  D - Display Database  I - Interpret

ALTER TABLESPACE : SSEMMDBC.SS13DPT          (Nonpartitioned          )

Buffer Pool . . . . BP16          Close Rule . . . YES Max Rows . . 255
Lock Size . . . . . ANY          Lock Max . . SYSTEM
Max Partitions . . . 0          LOG . . . . . YES

          Primary      Secondary  Free Pct  Com E T S
S  Part Quantity      Quantity   Page Free prs R M T VCAT      Stogroup
GBPCache
- -----> -----

          0          12          12    0    5 NO  N Y I DB0BD      SYSDEFLT CHANGED
***** END OF DB2 DATA *****
```

Figure 15-23 Object in pending state

Press F3 to return to the System Catalog panel and enter PDC on the Option line to view the pending change (Figure 15-24).

```
DB2 Admin ----- DB0B DB2 Pending Definition Changes----- Row 4 from 7
Command ==>                                           Scroll ==> PAGE

Commands: DIS  UTIL  DROP
Line commands:
  T - Tables  D - Database  X - Indexes  S - Table spaces  UTIL - Utilities
  DIS - Display object  DROP - Drop changes  SQ - Statement text  I -Interpret
  ? - Show all line commands

Select Name          Qualifier T S Keyword          Value          Timestamp
      SS13*          *          * *
-----> -----
      SS13DPT          SSEMMDBC  S 1 MAXPARTITIONS  20          2010-11-08-17
***** END OF DB2 DATA *****
```

Figure 15-24 SYSIBM.SYSPENDINGDDL table



General options

This chapter discusses general options of the DB2 Administration Tool:

- ▶ GEN
- ▶ Catalog navigation
- ▶ Performance queries

16.1 GEN

The GEN command is used to extract DDL for an object and its dependents. You can select which dependent objects are to be extracted.

You can use GEN as a line command against one object or as a primary command against a list of objects. GEN is an integral part of the DB2 Administration Tool:

- ▶ It can be used as a line command or a primary command.
- ▶ It can run in batch or online.
- ▶ It is used as a stand-alone command and by the MIG, DB2 Object Comparison Tool, and work statement list member cloning functions.

Consequently, it is important that GEN supports the latest release of DB2 as soon as possible.

16.1.1 GEN prompt to run SQL

One of the newer features or capabilities of GEN is the ability to execute the SQL right after it has been extracted. The SQL is not physically stored anywhere. The following parameters must be specified on the Generate SQL from DB2 Catalog panel to use this feature:

- ▶ Add to work statement list must be set to No.
- ▶ Execution mode must be set to TSO.
- ▶ Prompt to run SQL must be set to Yes.

Masking may be used (Figure 16-1).

```
ADB2GEN n ----- DB0B Generate SQL from DB2 catalog ----- 16:52
Option ==>

Generate SQL statements for tablespace SSEMDBB.SS12 > DB2 System: DB0B
                                                    DB2 SQL ID: ADMR1
                                                    More: - +

Target DB2 version . . . (Current DB2 version: 1015)
Use Masking . . . . . NO (Yes/No)
Generate catalog stats . NO (Yes/No/Only)
Target cat qualifier . > (Default is SYSIBM)
Include DB2 pending chgs NO (Yes/No/Alter/Only)

Output file and execution mode:
Add to work stmt list . NO (Yes/No)
Data set name . . . . .
Data set disposition . OLD (OLD, SHR, or MOD)
Execution mode . . . . . TSO (BATCH or TSO)
Commit statements per . (Db, tS, Tb, All, None. Default is All)
DB2 defaults handling . (Keep, or Remove. Default is Keep)
Prompt to run SQL . . . YES (Yes/No. For TSO mode and no WSL)

DB2 Command output file:
Data set name . . . . .
Data set disposition . OLD (OLD, SHR, or MOD)
```

Figure 16-1 GEN parameter panel

When you press Enter, the extracted SQL is displayed in a temporary TSO data set. You can scroll through the SQL to view or update the extracted DDL. Press F3 and the DB2 Administration Tool switches you to the Execute & Copy panel used by the DDL line command (Figure 16-2).

```
ADB2GEND ----- DB0B SQL - Execute & Copy ----- 17:00
Option ==>

Execute the generated SQL : Yes (Yes/No)
Copy the SQL to dataset . : >
Disposition . . . . . : (SHR, OLD or MOD)
The specified dataset will be created if it does not exist.
```

Figure 16-2 SQL Execute & Copy panel

Enter Yes next to the “Execute the generated SQL” keyword and press Enter.

A panel opens and informs you that the SQL will be executed upon leaving this panel. Press F3 to close the panel and open the Statement Execution Prompt panel (Figure 16-3).

```
ADB2PSTM ----- DB0B Statement Execution Prompt ----- 17:17
Option ==>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now (Add an A for all stmts. For example 1A - Execute all stmts):
  1 - Execute the statement
  2 - Edit the statement
  3 - Create a batch job with the statement
  4 - Add the statement to the work statement list
CAN - Cancel
Work statement list dsn ==> 'TEAM76.ALTER.WSL'
Work statement list name ==> TSALT      Action ==> A (Append or Replace)
                                           More:      +
Statement that is about to be executed (first 28 lines, more stmts pending):
CREATE TABLESPACE SS12DPT
  IN SSEMMDBB
  USING STOGROUP SYSDEFLT
  PRIQTY 12 SECQTY 12
  ERASE NO
  FREEPAGE 0 PCTFREE 5
  GBPCACHE CHANGED
  TRACKMOD YES
  LOGGED
```

Figure 16-3 GEN Extracted DDL in a queue

To execute all of the SQL statements in the queue, issue a 1A on the Option line of the Statement Execution Prompt panel.

16.1.2 GEN and pending changes

The “Generate SQL from DB2 Catalog” panel provides a new option for DB2 10 for z/OS support, “Include DB2 Pending Changes”. Valid options are Yes, No, Alter, and Only (Figure 16-4).

```
ADB2GEN n ----- DB0B Generate SQL from DB2 catalog ----- 18:16
Option ==>

Generate SQL statements for database SSEMMDBB          DB2 System: DB0B
                                                    DB2 SQL ID: ADMR1
                                                    More: - +
Storage group for TS . . . > Storage group for IX . . . >
Target DB2 version . . . (Current DB2 version: 1015)
Use Masking . . . . . NO (Yes/No)
Generate catalog stats . NO (Yes/No/Only)
Target cat qualifier . > (Default is SYSIBM)
Include DB2 pending chgs Yes (Yes/No/Alter/Only)

Output file and execution mode:
Add to work stmt list . NO (Yes/No)
Data set name . . . . .
Data set disposition . OLD (OLD, SHR, or MOD)
Execution mode . . . . . TSO (BATCH or TSO)
Commit statements per . (Db, tS, Tb, All, None. Default is All)
DB2 defaults handling . (Keep, or Remove. Default is Keep)
Prompt to run SQL . . . NO (Yes/No. For TSO mode and no WSL)

DB2 Command output file:
Data set name . . . . .
```

Figure 16-4 GEN parameter panel

When set to Yes, the DB2 pending changes are appended as messages to the bottom of the DDL extracted by GEN (Figure 16-5).

```
File Edit Edit_Settings Menu Utilities Compilers Test Help

ISREDDE2 SYS10314.T172331.RA000.ADMR1.R0100357 Columns 00001 00072
Command ==> Scroll ==> CSR
000161 -----
000162 -- Table space=SSEMMDBB.SS12DPT
000163 -----
000164 --
000165 -- ALTER TABLESPACE SSEMMDBB.SS12DPT
000166 -- MAXPARTITIONS 20
000167 -- BUFFERPOOL BP8K0;
000168 --
***** ***** Bottom of Data *****
```

Figure 16-5 GEN output: Include DB2 Pending Changes = Yes

When set to Alter, the DB2 pending changes are appended as executable DDL SQL statements at the bottom of the extracted DDL (Figure 16-6).

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

ISREDDE2  SYS10314.T182033.RA000.ADMR1.R0100389      Columns 00001 00072
Command ==>                                         Scroll ==> CSR
001584 -----
001585 -- Table space=SSEMMDBB.SS12DPT
001586 -----
001587 --
001588 ALTER TABLESPACE SSEMMDBB.SS12DPT
001589     MAXPARTITIONS 20
001590     BUFFERPOOL BP8K0;

```

Figure 16-6 GEN output: Include DB2 Pending Changes = Alter

When “Include DB2 Pending Chgs” is set to Only, only the DB2 pending changes are displayed in an executable format in the output. In other words, no other DDL is generated by GEN (Figure 16-7).

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

ISREDDE2  SYS10314.T182505.RA000.ADMR1.R0100393      Columns 00001 00072
Command ==>                                         Scroll ==> CSR
000024 -----
000025 -- Table space=SSEMMDBB.SS12DPT
000026 -----
000027 --
000028 ALTER TABLESPACE SSEMMDBB.SS12DPT
000029     MAXPARTITIONS 20
000030     BUFFERPOOL BP8K0;
000031 --

```

Figure 16-7 GEN output: Include DB2 Pending Changes = ONLY

16.1.3 GEN statement types

DB2 Administration Tool V10 supports some new SQL statement types that can be generated by GEN:

- ▶ CREATE MASK: Extracts column mask definitions. Valid values are Yes or No.
- ▶ CREATE PERMISSION: Extracts row permission definitions. Valid values are Yes or No.
- ▶ ALTER TABLE ACTIVATE CONTROL: Extracts the ALTER TABLES SQL to activate column and row columns.

Figure 16-8 show the statement options.

```

ADB2GEN n ----- DB0B Generate SQL from DB2 catalog ----- 18:09
Option ==>

Generate SQL statements for database SSEMMDBB                      DB2 System: DB0B
                                                                DB2 SQL ID: ADMR1
                                                                More:      +

SQL statement types to be generated from the DB2 catalog:
CREATE DATABASE . . . . Y (Y,N)  GRANT access ON DATABASE . . Y (Y,N,A,R)
CREATE TABLESPACE . . . . Y (Y,N)  GRANT access ON TABLESPACE . Y (Y,N,A,R)
CREATE TABLE . . . . . Y (Y,N)  GRANT access ON TABLE . . . . Y (Y,N,A,R)
CREATE VIEW . . . . . Y (Y,N,D)  GRANT access ON VIEW . . . . Y (Y,N,A,R)
CREATE INDEX . . . . . Y (Y,N)  ALTER TABLE ADD FOREIGN KEY. Y (Y,N,D)
CREATE SYNONYM . . . . . Y (Y,N)  LABEL ON . . . . . Y (Y,N)
CREATE ALIAS . . . . . Y (Y,N)  COMMENT ON . . . . . Y (Y,N)
CREATE TRIGGER . . . . . Y (Y,N,D)  REBIND PLAN/PACKAGE . . . . Y (Y,N,D)
CREATE MASK . . . . . Y (Y,N)  ALTER TABLE ACTIVATE CONTROL Y (Y,N)
CREATE PERMISSION . . . . Y (Y,N)
CREATE STORAGE GROUP . . Y (Y,N)  GRANT use OF STORAGE GROUP . Y (Y,N,A,R)

New names/values for generated SQL: (leave blank to use current values)
Object schema . . . . . > Run SQLID . . . . . ADMR1
Object grantor . . . . . >
Alloc TS size as . . . . DEFINED      (DEFINED, USED, or ALLOC)
Database name . . . . .

```

Figure 16-8 GEN SQL Statement Options

See Chapter 13, “Security” on page 357 for more details about these object types.

16.2 Catalog navigation

DB2 Administration Tool supports catalog navigation for the new DB2 10 for z/OS objects and data types. Some of the more notable ones are described below.

There is a new option on the System Catalog panel, PDC. The purpose of this option is to display a list of DB2 pending changes.

Note: The addition of this command to the list has moved the Selection Criteria filtering further down the panel. To see the full set of filtering criteria, press F8.

Figure 16-9 show the System Catalog panel.

```

DB2 Admin ----- DSNT System Catalog ----- 12:34
Option ==>

Object options:
  AO - Authorization options
  G - Storage groups
  D - Databases
  S - Table spaces
  T - Tables, views, and aliases
  V - Views
  A - Aliases
  Y - Synonyms
  X - Indexes
  C - Columns
  N - Constraints
  DS - Database structures
  P - Plans
  L - Collections
  K - Packages
  M - DBRMs
  H - Schemas
  E - User defined data types
  F - Functions
  O - Stored procedures
  J - Triggers
  Q - Sequences
  DSP - DS with plans and packages
PDC - DB2 pending definition changes

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name      ==>          > Grantor ==>          >
Owner     ==>          > Grantee ==>          >
In D/L/H ==>          > Switch Catalog Copy ==> N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)

```

Figure 16-9 System Catalog panel

There are some new options on the Authorization Options panel for displaying the DB2 10 for z/OS row permissions and column masks (Figure 16-10).

```

DB2 Admin ----- DSNT System Catalog ----- 12:39
Option ==>

                                     More:      +
Authorization options:                DB2 System: DSNT
  OO - Object options                  DB2 SQL ID: DBA104
  GA - Storage group auths             PA - Plan authorizations
  DA - Database authorizations         LA - Collection authorizations
  SA - Table space authorizations      KA - Package authorizations
  TA - Table authorizations            HA - Schema authorizations
  VA - View authorizations             EA - User defined data typeauthorizations
  CA - Column authorizations           FA - Function authorizations
  ZA - System authorizations           OA - Stored procedure authorizations
  UA - User authorizations             QA - Sequence authorizations
  RA - Resource authorizations         TR - Trusted contexts
  RO - Roles                           PM - Permissions
  CM - Column masks

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name      ==>                               > Grantor ==>                               >
Owner     ==>                               > Grantee ==>                               >
In D/L/H ==>                               > Switch Catalog Copy ==> N   (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)

```

Figure 16-10 Authorization options

Refer to 13.5.2, “Row permissions” on page 384 and 13.5.1, “Columns masks” on page 373 for more details about these two topics.

Another set of notable commands are the PLANMGMT / QUALIFIER options on the Packages panel. If you click the PLANMGMT keyword, the information displayed in the body of the panel shows the value stored in the PLANMGMT column of the SYSIBM.SYSPACKAGE table. If you click the QUALIFIER keyword, the PLANMGMT column is replaced by the QUALIFIER column (Figure 16-11 and Figure 16-12 on page 493).

```

DB2 Admin ----- DSNT Packages ----- Row 1 to 11 of 763
Command ==>                                           Scroll ==> PAGE

Commands: BIND REBIND FREE VERSIONS GRANT ALL PLANMGMT
Line commands:
DP - Depend A - Auth T - Tables V - Views X - Indexes S - Table spaces
Y - Synonyms RB - Rebind F - Free B - Bind BC - Bind copy GR - Grant
EN -Enab/disab con PL - Package lists P - Local plans LP - List PLAN_TABLE
I - Interpret SQ - SQL in package VE - Versions D - Databases RO - Role

          V I V O Qualifier R ED
S  Collection          Name      Owner      Bind Timestamp  D S A P L XR
  *              *          *          *              * * * * *
-----
DSNADM          DSNADMT0 KLTAYLO  2010-10-05-19.10 R U Y Y KLTAYLO  C N
DSNTIAP         DSNTIAP  KLTAYLO  2010-10-05-13.04 R   Y Y KLTAYLO   N
DSNESPSCS       DSNESM68 KLTAYLO  2010-10-05-13.04 R S Y Y KLTAYLO   N
DSNESP RR       DSNESM68 KLTAYLO  2010-10-05-13.04 R R Y Y KLTAYLO   N
DSNESPUR        DSNESM68 KLTAYLO  2010-10-05-13.04 R U Y Y KLTAYLO   N
DSNEDCL         DSNNECP68 KLTAYLO  2010-10-05-13.04 R S Y Y KLTAYLO   N
DSNUTIL         DSNUGSQL  KLTAYLO  2010-10-05-13.04 B S Y Y KLTAYLO   N
DSNADM          DSNADM GU KLTAYLO  2010-10-05-19.10 R S Y Y KLTAYLO  C N
DSNADM          DSNADM GW KLTAYLO  2010-10-05-19.10 R S Y Y KLTAYLO  C N
DSNADM          DSNADM IZ KLTAYLO  2010-10-05-19.10 R S Y Y KLTAYLO   N
DSNADM          DSNADM JS KLTAYLO  2010-10-05-19.10 R S Y Y KLTAYLO   N

```

Figure 16-11 Packages QUALIFIER

```

DB2 Admin ----- DSNT Packages ----- Row 1 to 11 of 773
Command ==> Scroll ==> PAGE

Commands: BIND REBIND FREE VERSIONS GRANT ALL QUALIFIER
Line commands:
DP - Depend A - Auth T - Tables V - Views X - Indexes S - Table spaces
Y - Synonyms RB - Rebind F - Free B - Bind BC - Bind copy GR - Grant
EN -Enab/disab con PL - Package lists P - Local plans LP - List PLAN_TABLE
I - Interpret SQ - SQL in package VE - Versions D - Databases RO - Role

S Collection Name Owner Bind Timestamp V I V O Plan R ED
* * * * * D S A P Mgmt L XR
* * * * * * * * * * * * * *

-----
DSNADM DSNADMT0 KLTAYLO 2010-10-05-19.10 R U Y Y E C N
DSNTIAP DSNTIAP KLTAYLO 2010-10-05-13.04 R Y Y E N
DSNESPSCS DSNESM68 KLTAYLO 2010-10-05-13.04 R S Y Y E N
DSNESP RR DSNESM68 KLTAYLO 2010-10-05-13.04 R R Y Y E N
DSNESPUR DSNESM68 KLTAYLO 2010-10-05-13.04 R U Y Y E N
DSNEDCL DSNNECP68 KLTAYLO 2010-10-05-13.04 R S Y Y E N
DSNUTIL DSNUGSQL KLTAYLO 2010-10-05-13.04 B S Y Y E N
DSNADM DSNADM GU KLTAYLO 2010-10-05-19.10 R S Y Y E C N
DSNADM DSNADM GW KLTAYLO 2010-10-05-19.10 R S Y Y E C N
DSNADM DSNADM IZ KLTAYLO 2010-10-05-19.10 R S Y Y E N
DSNADM DSNADM JS KLTAYLO 2010-10-05-19.10 R S Y Y E N

```

Figure 16-12 Packages PLANMGMT

16.3 Performance queries

DB2 Administration Tool provides a set of canned queries. These queries can be accessed by entering a 3 on the Option line of the DB2 Administration Tool Menu panel. A new query for Indexes not used in the last nn days has been added (Figure 16-13).

```
ADB23 min ----- DB2 Performance Queries ----- 16:41
Option ==>

                                     More:  -
  4 - Table spaces with more than 5  percent dropped space
  5 - Table spaces with locking size = 'S' (table space locking)
  6 - Index with 2  or more levels
  7 - Indexes with 150  or more leaf page distance
  8 - Indexes on tables with fewer than 6  pages
  9 - Indexes not used by any plan or package
 10 - Table spaces containing more than one table
 11 - Table spaces without SPACE information
 11X - Indexes without SPACE information
SPACE information is required for options 12 through 13
 12 - Table spaces exceeding allocated primary quantity
 12X - Indexes exceeding allocated primary quantity
 13 - Allocated and used space for table spaces
RTS Real-Time Statistics tables are required for options 14 and 14X
 14 - Table Space maintenance recommendations
 14X - Index Space maintenance recommendations
 15 - Indexes not used within 40  days
```

Figure 16-13 Performance Queries panel

This information is based on RUNSTAT data in the SYSINDEXES table. The indexes are shown on the Indexes panel (Figure 16-14).

ADB21X in ----- DB0B Indexes ----- Row 1 to 11 of 13
Command ==> Scroll ==> PAGE

Commands: DIS STA STO ALL
Line commands:
T - Tables D - Database G - Storage group P - Plans C - Columns
DIS - Display index space STA - Start index space STO - Stop index space
? - Show all line commands

Select Index Name		Index Schema	Table Name	Table Schema	U	Cols	C	C	CC
M	*	*	*	*	*	*	*	*	*
	XACT1	DSN81010	ACT	DSN81010	P	1	N	Y	NN
	XACT2	DSN81010	ACT	DSN81010	U	1	N	Y	NN
	XDEPT1	DSN81010	DEPT	DSN81010	P	1	N	Y	NN
	XDEPT2	DSN81010	DEPT	DSN81010	D	1	N	Y	NN
	XDEPT3	DSN81010	DEPT	DSN81010	D	1	N	Y	NN
	XEMP1	DSN81010	EMP	DSN81010	P	1	Y	Y	NN
	XEMP2	DSN81010	EMP	DSN81010	D	1	N	N	NN
	XEMPPROJACT1	DSN81010	EMPPROJACT	DSN81010	U	4	N	Y	NN
	XEMPPROJACT2	DSN81010	EMPPROJACT	DSN81010	D	1	N	Y	NN
	XPARTS	DSN81010	PARTS	DSN81010	D	1	N	N	NN
	XPROJ1	DSN81010	PROJ	DSN81010	P	1	N	Y	NN

Figure 16-14 List of Indexes: Unused



Part 7

Appendixes

This part contains the following appendix: Appendix A, “Reference tables” on page 499.



A

Reference tables

In this appendix, we provide details about the options and parameters referenced in the chapters.

The appendix contains the following topics:

- ▶ “DB2 Administration Tool install and upgrade planning worksheet” on page 500
- ▶ “Mask translation names” on page 504
- ▶ “Migration parameter details” on page 506

DB2 Administration Tool install and upgrade planning worksheet

This section of the appendix provides a worksheet (Table A-1) that can be used when you are planning to install or upgrade DB2 Administration Tool. Examples in Chapter 2, “Product setup” on page 13 use the contents of this worksheet to demonstrate a detailed example of installing the product in a virtual environment.

Table A-1 DB2 Administration Tool install and upgrade planning worksheet

Item No.	Item	Starting or default value (if applicable)	Chosen value
1.	Do I need to create copies of the original SMP/E target libraries?	No	
2.	High-level qualifier for DB2 Administration Tool operational libraries.	None	
3.	Prefix of last qualifier for the DB2 Administration Tool operational libraries.	SADB	
4.	High-level qualifier for DB2 Object Compare operational libraries.	None	
5.	Prefix of last qualifier for the DB2 Object Compare operational libraries.	SGOC	
6.	Am I going to have separate custom libraries?	No	
7.	High-level qualifier for the customization libraries (if answer for item 6 is Yes).	None	
8.	Prefix of last qualifier for the customization libraries (if answer for item 4 is Yes).	None	
9.	Is there a need for a variable-block CLIST and EXEC library?	No	
10.	DB2 subsystem name.	DSN	
11.	Release level of the DB2 subsystem.	None	
12.	DB2 runlib library name for the subsystem.	SYS1.DSN.RUNLIB.LOAD	
13.	DB2 exit library name for the subsystem.	SYS1.SDSNEXIT	
14.	DB2 load library name for the subsystem.	SYS1.SDSNLOAD	
15.	DB2 message library name for the subsystem.	None	
16.	DB2 panel library name for the subsystem.	None	
17.	DB2 CLIST library name for the subsystem.	None	
18.	DB2 skeleton library name for the subsystem.	None	
19.	DB2 table library name for the subsystem.	None	
20.	DB2 REXX EXEC library name for the subsystem.	None	
21.	DSNTEP2 plan name.	DSNTEP2	

Item No.	Item	Starting or default value (if applicable)	Chosen value
22.	DSNTIAD plan name.	DSNTIAD	
23.	ISPF Load Library.	None	
24.	Is the ISPF Load Library in the system LINKLIST?	None	
25.	ISPF Message Library.	None	
26.	ISPF Table Library.	None	
27.	Do I want to create a new storage group for my checkpoint database? (Yes/No)	Yes	
28.	Checkpoint database storage group name.	ADBGCH	
29.	Checkpoint database name.	ADBDCH	
30.	Table space name for the ADBCHKPT table.	ADBSCH	
31.	Table space name for the ADBPART table.	ADBSPART	
32.	Table space name for the ADBCHK table.	ADBSCHK	
33.	Table space name for the ADBHOLD table.	ADBSHOLD	
34.	Table space name for the ADBDBETLIST table.	ADBSBETL	
35.	User ID used to create the checkpoint database.	ADB	
36.	Schema name for the checkpoint database tables.	ADBCKTSC	
37.	Schema name for the checkpoint database indexes.	ADBCKXSC	
38.	Do I need to create a temporary database?	No	
39.	Do I need to create a segmented temporary table space? This item is only relevant if the answer to item 38 is Yes.	No	
40.	Segmented temporary table space database name. This item is only relevant if the answer to item 39 is Yes.	TEMPDB	
41.	Segmented temporary table space storage group name. This item is only relevant if the answer to item 39 is Yes.	SYSDEFLT	
42.	Segmented temporary table space name. This item is only relevant if the answer to item 39 is Yes.	TEMPTS	
43.	User ID used to create the temporary database and segmented temporary table space. This item is only relevant if the answer to item 39 is Yes.	ADB	
44.	Do I want to create a new storage group for my change management database?	Yes	
45.	Change management database storage group name.	ADBGCHG	
46.	Change management database name.	ADBDCHG	
47.	Table space name for the ADBCHG table.	ADBSCHG	
48.	Table space name for the ADBCHGS table.	ADBSCHGS	
49.	Table space name for the ADBCHGSR table.	ADBSCHGR	

Item No.	Item	Starting or default value (if applicable)	Chosen value
50.	Table space name for the ADBCPREREQ table.	ADBSCPRQ	
51.	Table space name for the ADBCMASK table.	ADBSCMSK	
52.	Table space name for the ADBCMASKS table.	ADBSCMSS	
53.	Table space name for the ADBCIGNORE table.	ADBSCIGN	
54.	Table space name for the ADBCIGNORES table.	ADBSCIGS	
55.	Table space name for the ADBCVERSION table.	ADBSCVER	
56.	Table space name for the ADBCVERLINES table.	ADBSCVLN	
57.	Table space name for the ADBCVERSCOPE table.	ADBSCVSC	
58.	Table space name for the ADBCVERSCOPES table.	ADBSCVSS	
59.	Table space name for the ADBCID table.	ADBSCID	
60.	User ID used to create the change management database.	ADB	
61.	Schema name for the change management database tables.	ADBCMTSC	
62.	Schema name for the change management database indexes.	ADBCMASC	
63.	Would you like users to be able to delete cancelled changes from the change management database?	No	
64.	Main Product Plan Name.	ADB	
65.	Main Product Collection Name.	ADBL	
66.	SMP/E Data Set Volume Serial.	VLSRNM	
67.	SMP/E Volume Unit Type.	SYSALLDA	
68.	Do I want to invoke DB2 Administration Tool as another user?	No	
69.	Under what user ID do I want to invoke DB2 Administration Tool? This item is only relevant if answer to item 68 is Yes.	None	
70.	Unit name for batch work data sets.		
71.	Unit name for the TSO work data sets.		
72.	Would you like users to have the option to operate against copies of the DB2 catalog?	No	
73.	Do I want to create a new storage group for my catalog copy version database? This time is only relevant if the answer to item 72 is Yes.	Yes	
74.	Catalog copy version database storage group name. This time is only relevant if the answer to item 72 is Yes.	ADBGCC	
75.	Catalog copy version database name. This item is only relevant if the answer to item 72 is Yes.	ADBDC	

Item No.	Item	Starting or default value (if applicable)	Chosen value
76.	Table space name for the ADBCATVT table. This item is only relevant if the answer to item 72 is Yes.	ADBSCC	
77.	User ID used to create the catalog copy version database. This item is only relevant if the answer to items 72 is Yes.	ADB	
78.	Schema name for the catalog copy version database tables. This item is only relevant if the answer to item 72 is Yes.	ADBCCTSC	
79.	Schema name for the catalog copy version database indexes. This item is only relevant if the answer to item 72 is Yes.	ADBCCXSC	
80.	Workload Manager (WLM) environment name.	DSNWLM1	
81.	Schema name for the stored procedure to execute DB2 commands when connected to a remote site.	ADB	
82.	RUNSTATS view schema name.	ADBRVSC	
83.	DB2 Security Exit Type.	STD	
84.	System Identification Method.	None	
85.	Installation Name.	None	
86.	Node Name.	None	
87.	Will the product be run on a JES3 system?	No	
88.	UNICODE Translation Technique.	None	
89.	DB2 Admin APF library.	None	
90.	Job parameter SYSAFF for DB2 batch utility jobs.	None	
91.	Job class to be used for DB2 batch utility jobs.	A	
92.	DB2 started task name.	None	
93.	DB2 group name.	None	
94.	DB2 utility data set prefix.	<userid>	
95.	DB2 remote subsystem name.	None	
96.	DB2 remote subsystem location.	None	
97.	Authorization switching enabled?	No	
98.	ISPF application ID.	None	
99.	Prompt options.	None	
100.	Reset to default at startup.	None	
101.	DSNUPROC step count.	1	
102.	SSID switch enabled?	Yes	
103.	DB2 High Performance Unload enabled?	No	

Item No.	Item	Starting or default value (if applicable)	Chosen value
104.	DB2 High Performance Unload load library.	None	
105.	DB2 High Performance Unload panel library.	None	
106.	REXX user exit libraries.	None	

Mask translation names

This section provides the list of properties that can be changed by the Migrate and Compare features (described in Chapter 8, “The MIG line command” on page 213) when using the mask capabilities. Refer to “Creating masks using a data set” on page 216 for more details.

The full list of mask translations is in Table A-2. Note that DBROLE, TSROLE, TBROLE, and IXROLE are not currently used.

Table A-2 Translation mask names

Name	Parent	Grandparent	Description
COLNAME			Column name.
		NAME	All below.
COLLNAME		NAME	Collection name.
DBNAME		NAME	Database name.
DBRMNAME		NAME	DBRM name. ^a
GBPNAME		NAME	Group buffer pool name. ^a
GRPNAME		NAME	Group name.
IXNAME		NAME	Index name.
PGMNAME		NAME	Program name; synonym for DBRM.
PKNAME		NAME	Package name.
PLNNAME		NAME	Plan name.
SFNAME		NAME	Structure facility name. ^a
STPNAME		NAME	Stored procedure name.
TBNAME		NAME	Tables, alias, synonym, and view name.
TGNAME		NAME	Trigger name.
TSNAME		NAME	Table space name.
UDFNAME		NAME	User-defined function name.
UDTNAME		NAME	User-defined function name.
VCATNAME		NAME	VCAT name.
	SGNAME	NAME	All storage group names.
IXSGNAME	SGNAME	NAME	Storage group name for indexes.

Name	Parent	Grandparent	Description
TSSGNAME	SGNAME	NAME	Storage group name for table spaces.
	BPNAME	NAME	All buffer pool names.
IXBPNAME	BPNAME	NAME	Buffer pool names for indexes.
IXTSNAME	BPNAME	NAME	Buffer pool names for table spaces.
PMNAME		NAME	Masks the name of the row permissions.
MKNAME		NAME	Masks the name of the column mask.
GRANTEE	GRANTID	AUTHID	Grantee.
GRANTOR	GRANTID	AUTHID	Grantor.
OWNER		AUTHID	Owner, creator, and so on. Masks the OWNER field.
DBOWNER	OWNER	AUTHID	Database owner. ^b
IXOWNER	OWNER	AUTHID	Owner of the index. ^c
TBOWNER	OWNER	AUTHID	Owner of the index. ^c
TSOWNER	OWNER	AUTHID	Owner of table space. ^b
SCHEMA		AUTHID	Used to mask SCHEMA field.
TBSCHEMA	SCHEMA	AUTHID	Masks the table creator field. ^c
IXSCHEMA	SCHEMA	AUTHID	Masks the index creator field. ^c
XMLSCHID			Masks the registered XML schema name in an XML - type modifier.
PMSCHHEMA	SCHEMA	AUTHID	Masks the schema of the row.
MKSCHEMA	SCHEMA	AUTHID	Masks the schema of the column mask.
SQLID		AUTHID	Current SQLID. ^a
COMPRESS			Whether a table space or table space partition is compressed.
SEGSIZE			Segment size.
DSSIZE			Maximum size in gigabytes for each partition for a partitioned table space.
	PRIQTY		Minimum primary space allocation for a DB2 data set.
IXPRIQTY	PRIQTY		Minimum primary space allocation for an index space.
TSPRIQTY	PRIQTY		Minimum primary space allocation for an index space.
	SECQTY		Minimum secondary space allocation for a DB2 data set.
IXSECQTY	SECQTY		Minimum secondary space allocation for an index space.
TSSECQTY	SECQ?TY		Minimum primary space allocation for a table space.
DEFER			Whether to build the index at CREATE time.
	DEFINE		Whether the underlying data sets for the table space or index space are created when the object is created or when data is inserted into the object.

Name	Parent	Grandparent	Description
IXDEFINE	DEFINE		Whether the underlying data sets for the index space are created when the object is created or when data is inserted into the object.
TSDEFINE	DEFINE		Whether the underlying data sets for the table space are created when the object is created or when data is inserted into the object.
TCNAME		NAME	Masks a trusted context name.
ROLE	AUTHID		Masks a role name.
DBROLE	ROLE	AUTHID	Masks a role associated with a database.
TSROLE	ROLE	AUTHID	Masks a role associated with a table space.
TBROLE	ROLE	AUTHID	Masks a role associated with a table role.
IXROLE	ROLE	AUTHID	Masks a role associated with an index.
HASHSPC			Overwrites Hash Space Integer.
TBINLOBL			Overwrites INLINE LENGTH integer value for tables.
DTINLOBL			Overwrites INLINE LENGTH integer value for distinct types.

a. Not used when cloning Work Statement Lists

b. Not used when cloning Work Statement Lists

c. The OWNER of the object in DB2 8, and the SCHEMA of the object in DB2 9 for z/OS.

Migration parameter details

This section of the appendix provides a detailed description of the parameters used in the Migration Parameter panel shown in Figure 8-16 on page 225. The parameters used in the Migration Parameter panel are shown in Table A-3.

Table A-3 Migration parameter details

Parameter name	Values	Description
Worklist Name		Used for WSL name and in data set name.
PDS for jobs		PDS for generated jobs.
Prefix for data sets		This prefix must be the prefix that you are allowed to allocate.
DB2 Subsystem ID (SSID) ^a		SSID of source DB2 subsystem.
DB2 Release ^a	Valid DB2 Version, Release Modification number	DB2 release number. DB2 Administration Tool removes clauses from DDL that are not valid for the specified release.
Target System node name ^a	Valid DB2 Version, Release Modification number	Target node name. Must be specified if Submit Job is Yes.
Submit job at local ^a	Y/N	Generates jobs for target system in such a way that they can be submitted at the local site. If No is set, then the jobs have to be transferred to the target system.

Parameter name	Values	Description
DB2 sample pgm library ^a		DB2 sample pgm at target.
Use Cust Table Settings	Y/N	Use values defined at customization time or use user defined libraries specified on the panel. This parameter allows the default libraries to be used without having to erase the library fields.
DB2 Admin APF Library ^a		User defined library, used if Cust Table = N. If Cust table = N and this field is blank, the customization library is used.
DB2 exit library ^a		User defined library, used if Cust Table = N. If Cust table = N and this field is blank, the customization library is used.
DB2 load library ^a		User defined library, used if Cust Table = N. If Cust table = N and this field is blank, the customization library is used.
New TS storage group		Target storage group for table space.
New IX storage group		Target storage group for index space.
New Database		Target database name. <ul style="list-style-type: none"> ► If migrating at the database level, the database is created for explicit objects. ► If migrating at the table space level, it is assumed that the database exists at the target. ► If migrating at the table level, it is assumed that the database and table space already exist at the target.
New schema of objects		Target schema.
New grantor		Target grantor.
Catalog qualifier		Target catalog qualifier. Used in catalog statistics DML. This parameter is only used if you are migrating catalog statistics.
Generate MIG jobs in batch	Y/N	Batch or online generate.
Generate Work statement list		Work statement list (WSL) or migrate commands.
Use masking for batch migrate	Yes/No	Use masking parameters for migration. This parameter cannot be used if WSL is used. If WSL is used, then the mask can be used from the WSL panels.
Combine job steps	Yes/No	One job or several jobs are generated. If Y is set, all the steps are combined into three jobs. The first and second jobs are executed on the source and the third on the target system. If more than 254 job steps are required, then multiple jobs are generated for that group.
Member prefix		This variable is used to build the job names, up to 5 characters.

Parameter name	Values	Description
Scope of migrate	DDL/Data/Catalog Statistics	The scope of the migration. At least one parameter needs to be chosen. Defaults to DDL and Data.
DROP on target before CREATE	Y/N	Drops the object on the target before creating new objects. Storage groups are not dropped. This parameter is only valid if DDL selected. If the target object has RESTRICT on DROP, then you have to remove this parameter before running the DROP command.
Create storage group		A storage group is created at the target. The attributes of the new storage group are gained from the source subsystem.
Generate GRANT statements		Controls if GRANT statements.
RUN SQLID		SQLID is used during execution. If NONE is set, then no SET CURRENT SQLIDs are generated.
Unload Method		Determines whether to use the Unload utility or the IBM High Performance unload product. The option for HPU does not show if the product is not installed.
Parallel utilities	Y/N	Runs utilities in parallel. If selected, the utilities are in multiple jobs to allow them to be scheduled to run in parallel instead of serially.
RUN CHECK DATA	Y/N	Generates a CHECK DATA utility step for the explicitly defined migrated table spaces.
Run RUNSTATS	Y/N	Generates a RUNSTATS utility step for the explicitly defined migrated table spaces.
Run IMAGE COPY	Y/N	Generates an IMAGE COPY utility step for the explicitly defined migrated table spaces.
Run REBIND	Y/N	Generates a REBIND step for the target system.
Generate Template statements	Y/N	Determines whether you want to use templates or not.
Use customized utility options	Y/N	Determines whether you want to use customized utility options.

- a. This field is ignored for an add migrate request to the work statement list. The line command R on the ADB2W1 panel supplies a work statement list. The line command R on the ADB2W1 panel supplies this information when building the job on the target system.

Abbreviations and acronyms

AIX®	Advanced Interactive eXecutive	DSC	dynamic statement cache (local or global)
ALC	Admin ALTER	DSN	data set name
APAR	authorized program analysis report	DTT	declared temporary tables
APF	Authorized Program Facility	EA	extended addressability
ARM	automatic restart manager	EBCDIC	Extended Binary Coded Decimal Interchange Code
ASCII	American National Standard Code for Information Interchange	ECS	enhanced catalog sharing
BLOB	binary large objects	ECSA	extended common storage area
CCA	client configuration assistant	EDM	environment descriptor management
CCSID	coded character set identifier	EOF	end of file
CD	compact disk	ERP	enterprise resource planning
CF	coupling facility	ESA	Enterprise Systems Architecture
CFCC	coupling facility control code	ESP	Enterprise Solution Package
CFRM	coupling facility resource management	ETR	external throughput rate
CLI	call level interface	FTD	functional track directory
CLP	command-line processor	FTP	File Transfer Program
CM	Change Management	GB	gigabyte (1,073,741,824 bytes)
CM	Create mask	GBP	group buffer pool
CPC	central processor complex	GEN	generate
CPU	central processing unit	GLOBAL	global zone
CSA	common storage area	GRS	global resource serialization
CTT	created temporary table	GUI	graphical user interface
DASD	direct access storage device	HPJ	high performance Java™
DB2 PE	DB2 Performance Expert	HPU	High Performance Unload
DBA	database administrator	I/O	input/output
DBAT	database access thread	IBM	International Business Machines Corporation
DBD	database descriptor	ICF	integrated catalog facility
DBID	database identifier	ICF	integrated coupling facility
DBRM	database request module	ICMF	internal coupling migration facility
DCL	data control language	IFCID	instrumentation facility component identifier
DDCS	distributed database connection services	IFI	instrumentation facility interface
DDF	distributed data facility	IPLA	IBM Program Licence Agreement
DDL	data definition language	IRLM	internal resource lock manager
DLL	dynamic load library manipulation language	ISPF	interactive system productivity facility
DML	data manipulation language	ISV	independent software vendor
DNS	domain name server	IT	information technology
DRDA	distributed relational database architecture	ITR	internal throughput rate

ITSO	International Technical Support Organization	SMIT	System Management Interface Tool
IVP	installation verification process	SSID	subsystem ID
JDBC	Java Database Connectivity	SU	service unit
JFS	journaled file systems	TARG	target zone
JNDI	Java Naming and Directory Interface	UOW	unit of work
JVM	Java Virtual Machine	VSAM	Virtual Storage Access Method
KB	kilobyte (1,024 bytes)	WSL	work statement list
LOB	large object		
LPAR	logical partition		
LPL	logical page list		
LRECL	logical record length		
LRSN	log record sequence number		
LUW	logical unit of work		
LVM	logical volume manager		
MB	megabyte (1,048,576 bytes)		
MIG	Migrate		
NPI	non-partitioning index		
OC	Object Compare		
ODB	object descriptor in DBD		
ODBC	Open Data Base Connectivity		
OS/390	Operating System/390®		
PAV	parallel access volume		
PDS	partitioned data set		
PIB	parallel index build		
PSID	pageset identifier		
PSP	preventive service planning		
PTF	program temporary fix		
QA	Quality Assurance		
QMF™	Query Management Facility™		
RACF®	Resource Access Control Facility		
RBA	relative byte address		
RDEF	redefine an index		
RECFM	record format		
RI	referential integrity		
RID	record identifier		
RR	repeatable read		
RRS	resource recovery services		
RRSAF	resource recovery services attach facility		
RS	read stability		
RST	restart line command		
RUN	running		
SDK	software developers kit		

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 511. Note that some of the documents referenced here might be available in softcopy only.

- ▶ *DB2 10 for z/OS Technical Overview*, SG24-7892
- ▶ *DB2 for z/OS Administration Tools for Enhanced Change Management*, SG24-7441
- ▶ *Extremely pureXML in DB2 10 for z/OS*, SG24-7915

Other publications

These publications are also relevant as further information sources:

- ▶ *IBM DB2 Administration Tool for z/OS Version 10 Release 1 User's Guide and Reference*, SC19-3033
- ▶ *IBM DB2 Object Comparison Tool for z/OS Version 10 Release 1 User's Guide*, SC19-3037
- ▶ *ISPF User's Guide Volume II*, SC34-4823

Online resources

These websites are also relevant as further information sources:

- ▶ DB2 10 for z/OS
<http://www.ibm.com/software/data/db2/zos/>
- ▶ DB2 Tools for z/OS
<http://www.ibm.com/software/data/db2imstools/products/db2-zos-tools.html>

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

ADB 35, 66, 221, 240, 254, 266, 294
ADB2 dmin 331
ADB2 dmin menu 83, 112
ADB2 panel 129
ADB21 min 332, 342, 369, 467
ADB21 min menu 103, 113
ADB21OI1 panel 105
ADB21PM n 380
ADB21TA n 336, 340, 389, 445
ADB22 min 295
ADB26CO n 85
ADB27CA n 119, 206
ADB2A2WL Exec 264
ADB2ATH 57
ADB2C2L n 219
ADB2CUST 48, 66, 284, 460
ADB2DB2D 283
ADB2RCM 62
ADB2RCPC 63
ADB2UCUS skeleton 27
ADB2UTIL 57
ADBALTER JCL
 job 170
 job stream 211
ADBBIND 44–45
ADBCATVT 17, 58
ADBCHG 16
ADBCHGS 16
ADBCHKPT 15, 21
ADBCVERSCOPE 16
ADBDIAG 124
ADBL 25, 66, 219
ADBL CLIST 47–48
ADBP6CTB panel 325
ADBTEP2 14–15, 71, 75, 371
ADD 91–92, 110, 143–144, 188–189, 223–224,
264–265, 335, 379, 419, 489
ADDFK 119–120, 176, 178–179, 427
additional indexes 14
additional views 14
ADDRI 223
ADMR1.SS12 DPT3 254
ADMR1.SS12 EMP2 254
ADMR1.SS13 DPT3 254
ADMR1.SS13 EMP2 254
ADMR1D Job 254
ADMR2.MEMB ER_LIST 90
ADMR2.TBAD D304 125
ADMR3.CUST OMER 375
 MASK ADMR3.INCOME_BRANCH 381
ADMR3.TD76 TB02_EMP 228
 INDEX TD76XB02 228
ADMR3.TD76 TB06_EMPPROJACT 228
advisory reorg 466
AL 73, 91–92, 150, 166, 306, 378, 418, 466
 command 166, 335, 401, 432, 445
ALC 115, 166, 263, 276
Aliases panel 131, 332, 343
ALT
 command 115, 166, 353, 437–438
 processing 119
 central hub 136
 usage 306
ALTER 5, 74, 82, 88, 110, 119, 144, 166, 232, 260,
336–337, 368, 419, 466, 486
 command 120, 150, 190
ALTER BUFFERPOOL 461, 464
 new option 461
alter line
 commands 166
alter option 70, 119, 176, 427, 432
Alter Related
 panel 181
 Table panel 191
ALTER statement 182, 405, 420, 429, 466
ALTER stmt 96, 152, 338–339
ALTER TABLE 110, 150–151, 337–338, 379, 441, 488
 DNET305.TIME Z3 459
 DROP VERSIONING statement 354
 TEAM83.TD83 TB11_PARTTB 454–455
ALTER Table 150, 168, 188, 335–336, 379, 438, 489
Alter Table Add Clone panel 151
Alter Tables panel 119, 171, 427
 appropriate information 176
 Command line 171
 full table name 193
ALTER TABLESPACE 468
 SSEMMDBB.SS12 DPT 469, 487
 SSEMMDBC.SS13 DPT 480
ANALYZE 123, 167, 174, 256
AND 22, 90, 95, 123, 156, 254, 277, 296, 350, 371, 455
And/or other selection criterion 103, 113, 145, 299, 332,
342, 349, 361, 467, 490
AO 103, 113, 299, 332, 361, 467, 490
AP 410
APPLY 121, 167–168, 236, 260, 428
AT 264
Attributes 395
AUDIT None 333–334, 411, 447
audit policy 358, 411
AUTH 67–68
Auth ID 105, 115, 382
AUTHID 217, 294, 360

B
B_2 column 324, 326
 R line command 326

- BALANCE 117, 133
- base table 125, 144, 146–147, 159, 212, 306, 346, 351, 434
 - exact physical replica 144
- batch job 5, 120, 151, 155, 168, 229, 261, 286, 364, 420, 454, 468, 486
- batch parameters 75
- Bind copy (BC) 106, 492
- BIND PACKAGE 98
- bi-temporal table 305–306
- bl2llib 69
- BPNAME 217
- BPool 147, 158, 223, 287
- BR 157
- BROWSE 256, 351–352
- browse 157
- buffer pool 14, 115, 182, 419, 423, 466
 - BP15 479
 - name 128, 464
 - page size 23
 - size 16
- BUFFERPOOL BP16 426, 433, 476
- BUFFERPOOL BP8K0 452, 469, 487
- BUS_END column 318, 327
 - period column 328
- Business Time 203, 306, 423–424
- BUSINESS_TIME period 306
 - End column 323
 - Ending time column 318
 - Starting and Ending time column 324, 328
 - time column 318

C

- CAN 151, 261, 364, 420, 468, 486
- catalog copy version database 16–17
 - multiple copies 29
 - new copy 30
- catalog navigation 145
- catalog statistics 215, 225
- CCSID 115, 182, 323
- CCSID EBCDIC 123
- CHANGE 123, 255–256, 466
- Change DB2 System Checkpoint Frequency panel 453
- Change Management 5, 14, 68, 77, 83, 112, 216, 219, 237, 260, 283, 331, 337–339, 462
- change management database 14, 16
- changes xxv, 4–5, 14–15, 67, 71, 93, 103, 120, 125, 150, 152, 166, 213–214, 236, 238, 240, 260, 265–266, 299–300, 332, 336, 379, 426, 487
- character name 184, 251
- CHECK constraint 150, 336, 379, 445
- CHECK DATA 120, 159, 172, 184, 225, 253, 260, 428
- checkpoint database 14, 38
 - multiple copies 21
 - new copy 22
- chosen value 35
- CLASS 75, 123, 138, 254
- Clone field 115, 128, 149, 230–231, 263, 276
- clone name 231
- CLONE table 111, 128, 143–144, 147, 352, 411

- Clone Work Statement List panel 231, 266
- cloning 231, 259
- CLOSE
 - NO 123, 205, 256, 452
 - YES 123, 433, 476
- cmown 68
- CO 84, 282, 307
- COLLNAME 217
- column access 358, 383
- column mask 369, 373, 389, 491
 - ACT command 379
- Column Mask List panel 374
- Column MGR_NO 254
- column name 190, 309
- Column Name field 288, 309, 311, 419–420
- Column name field 382
- column T 127
- column type
 - TIMESTAMP 312, 337
 - TIMESTZ 459–460
- Columns 00001 90, 95, 123, 205, 254, 265, 426, 430–431, 476
- command
 - ALTER 120, 135, 152, 171, 427, 472
 - CONTINUE 135, 171, 198, 425
 - DDL 205, 426
 - DS 149
- command against a table (CT) 434
- Command line 4, 96, 117, 146, 148, 152, 157, 168, 189, 223, 230, 238, 295–296, 322, 325, 419, 425, 472
 - ALTER command 209
 - CM command 239
 - WSL command 429
- Compare
 - Masks 248
 - masks 218, 248
 - source 214
 - target 214, 247
- Comparison results 125, 170
- CONSTRAINT CUST_ID 334, 344
- Context IDs 397
- CONTINUE 117, 182, 188–189, 270–271, 423
- copy data 159–161
- CP 239
- CRE 395, 434
- Create Procedure panel 84
- CREATE stmt executed 90, 330
- Create Table (CT) 306
- Create Table Columns panel 309, 312, 434
- CREATE VIEW 191, 489
- CREATE_ID column 319
 - Operation type 321
- Create/drop/label/comment on objects field 295
- Created in DB2 Version field 115, 128, 469
- CRETAB 443
- CREX 448
- CS 84, 307
- CS - Table (CT) 84, 373
- CSR11 Cursor 90
- customization 20, 66–67, 269, 277

- DB2 Administration Tool 67, 69
- customization library 20
 - high-level qualifier 35, 38, 42, 44, 46–47, 58–62
- customization menu 66

D

- DACT option 383
- data access 358, 389
- DATA CAPTURE 150, 153, 323, 333–334, 379, 445, 447
- data conversion 214
- data length 309, 314, 439, 444
- data set 67, 110, 115, 149, 157, 172–173, 227, 252, 260, 287, 294, 418, 428
 - existing content 428
- data set names (DSN) 151, 155
- data type 82, 85, 113, 130, 145, 299–300, 309–310, 332, 341, 444, 450, 467, 489
- database 14, 22, 119, 166, 214, 229, 238, 282
 - change 215
 - DBADM authority 361
 - entity 406
 - environment 213
 - first query 223
 - integrity 378
 - level 222
 - object 16, 384
 - partial name 224
 - schema 448
 - valid line command 171
- DB 166–167, 408
- DB Name column 383
- DB0B Alter
 - Related 190
 - Table 150–151, 176, 188–189, 445–447
 - Table Space 470
- DB0BT.SDSN Load 254
- DB1S Table 146
- DB2
 - catalog 4, 6–8, 14, 16, 31–32, 60, 73, 82–83, 100, 112, 129, 169, 214, 231, 236, 244, 246, 266, 268, 283–284, 286–287, 331, 381, 383, 462, 470, 484–485
 - additional indexes 61
 - local copy 16
 - multiple copies 16
 - physical change 212
 - SELECT authority 60
 - catalog copy 14
 - version maintenance 101
 - data 91–92, 96, 113–114, 116–117, 146, 219, 241, 282, 309, 311–312, 362, 419, 421, 467
 - group 68
 - object 3, 119, 166, 213–214, 236, 275, 294
 - definition 8
 - IBMT00L.V10GOC00 35
 - identical set 214
 - performance 112, 239, 283, 331, 462
 - product 83, 112, 239, 283, 290, 331, 462
 - subsystem 5, 18, 20, 22, 66, 100, 213, 236, 282, 361, 410
 - DB2 group name 68

- level 48
- name 38
- parameter 66
- task name 68
- target catalog 252
- target subsystem 238
- utility 4–5, 68, 143, 232, 254, 260, 267
 - job 4
 - statement 265
- DB2 10 for z/OS 3, 37, 110, 218, 305–306, 357, 418, 434, 442–443, 466
 - ALTER changes 466
 - new function mode 306
 - new SQL data type 455
 - String delimiter 393
 - XML functions 444
- DB2 9 for z/OS 79, 109, 143, 357, 443
- DB2 Admin Alter ALT function 165
- DB2 Admin menu 261
- DB2 Administration Tool 4, 8, 13, 18–19, 21, 33–34, 57, 66–67, 81–82, 144–145, 213, 239, 281–282, 286, 312, 331, 361, 418–419, 465, 483
 - Change DB2 Admin Options panel 70
 - customization 68
 - evolution 8
 - General Parameters panel 66
 - main menu 77, 219, 223
 - overview 4
 - parameters 68, 269, 277
 - PR 76
 - V10 321, 352
 - AL command 335
 - features 340, 346, 354
 - work statement list 5
 - work statement list (WSL) 271
- DB2 BIND 283
- DB2 DATA 176
- DB2 Launchpad 25
- DB2 object
 - management 359
- DB2 Object Comparison Tool 3, 6, 13, 56, 83, 100, 112, 129, 235–236, 260, 284, 331, 462, 484
 - capability 236
 - main menu 243, 245
 - overview 6
 - V7.2 238
 - work statement list (WSL) 262
- DB2 Object Comparison Tool menu 243
- DB2 performance queries field 83
- DB2 SQL ID field 70, 83, 221, 282, 307, 331, 360, 453, 462, 485
- DB2 START 6
- DB2 start field 411
- DB2 Subsystem Parameters panel 67–68
- DB2 System Administration 462
- DB2 system administration field 83, 112, 239, 283, 331
- DB2 system administration menu 129
- DB2 System Administration panel 461
- DB2 system catalog field 83, 331
- DB2 System field 70, 160, 221, 262, 282, 307, 360, 453,

- 467, 485
- DBAADMR2.TPAA DMR2 115
- DBADMIN category 413
- DBAUSERD Job 75
- DBCLOB 337–338, 433, 439
- DBCS CCSID 115
- DBD Length 162, 474
- DBNAME 22, 217–218, 296, 301
- DBRMNAME 217
- DCL 7, 232, 267
- DD 83–84, 112, 123, 239, 243, 254, 268, 282–284, 307, 331, 371, 456
- DD DISP 254
- DDL 7–8, 21, 119–120, 144, 171, 176, 178–179, 214–215, 236–237, 267, 332, 359, 373, 426–427, 434, 475, 484–485
 - definition 238
 - file 8, 244, 246
- debug 82
- DEBUG MODE 88
 - valid options 88
- default ADBMG 225
- default value 35, 68, 89, 118, 309, 439, 444
- DEFAULT X 319
- DEFER 74, 89, 94, 168, 217–218, 253, 255, 440
- definition change 130, 342, 349, 467, 490
- Delta Version File 238
- DEPT table 187, 192
- DIS Thread 293
- DISPLAY Database 158, 162, 470
- Distributed DB2 systems field 83, 112, 239
- Distributed DB2 Systems menu 282
- distributed thds (DT) 453, 463
- DROP CLONE
 - option 153
 - SQL syntax 144, 155
 - syntax 154–155
 - Table 153–154
- DROP FOREIGN Key 150, 153, 336, 340, 379, 445, 457
- drop impact 404
- DROP VERSIONING 335
- DS 84, 103, 113, 130, 145, 149, 299–300, 307, 332, 453, 467, 490
- DS - Table (DT) 84, 307, 373
- DSN 5, 38–39, 76, 144, 149, 186, 216, 218, 224, 248, 254, 264–265, 371, 442
- DSNT415I SQLERRP 430–431, 480
- DSNT416I SQLERRD 430, 480
- DSNT418I SQLSTATE 430–431, 480
- DSNTEP2 job 21
- DSNTIAD 43, 45, 59–61
- DSNZPARM 360, 434–435
- DSNZPARM setting 32
- DT - Start (DS) 463
- dynamic SQL
 - Cache 460
 - statement 4, 460

EA 361,491

F

G

H

1

- IMPORT 256
- index source 125, 212, 254
- INDEX TEAM76
 - TD76XA06 203
- INDEX TEAM89 423–424
- Inline length 337, 434–435
- INLINE LOB 111
- inline LOB 434–435
 - column 434
 - DB2 Administration Tool V10 support 433
 - length 436
 - size 437
- inline portion 435, 437
- INS 220, 241, 410
- Insert command 188, 219
- interactive system productivity facility (ISPF) 6, 66, 296, 333, 375
 - Message Library 67
 - panel 15, 69, 166, 218
 - session 138
 - table 4, 66
 - Table Library 67
- Interpret option 429
- Interpret Work Statement List Options panel 232
- intrusive type 166
- IP address 373
- ISPF See *interactive system productivity facility*
- IXBPNAME 217
- IXNAME 217, 250
- IXSGNAME 217

J

- JCL Member 15, 185
- job card 39, 70
- Job Control Language (JCL) 14, 71, 120, 166, 184, 214, 288
- job name 185, 227, 399
- Job panel 120, 172, 428–429
- JOBPARM 75, 254

L

- Lang Parms 91–92
- LANGUAGE SQL 86
- LIKE 86–87, 113, 130, 145, 221, 240–241, 296, 299–300, 308, 332, 361, 467, 490
- LIKE operator 103, 342, 369, 477, 491
- line command 4, 59, 73, 91, 98, 104, 114, 127, 146–147, 165–166, 213, 238, 264, 279, 306, 323, 325, 362, 382, 418, 466, 468, 484
- LISTC 157
- LISTCAT 158
- Local plans (LP) 106, 492
- Lock max 115, 470
- Lock size 115, 182, 470
- Log DIAG 75

M

- MAKEPBG command 109

- MAKEPBR command 109
- Manage versions option 77
- Manage Work Statement Lists panel 230, 262, 276
- Manage/Restore dialogue 243
- mask definition 216, 248
- Mask Line 219, 249
- mask option 249
- mask panel 231
- masking 7, 215, 218, 231, 266, 373, 392, 485
- masks 77, 215, 239, 361, 491
- Max Partition 114, 182, 470
- Max Row 115, 182, 470
- Max rows/page 128
- MAXPARTITIONS 20 471, 487
- Member Cluster 117, 166, 466
- Member name 121, 159, 172, 216, 251, 260, 428
- middle qualifier 121, 172, 225, 251, 260, 428, 441
- MIG 76, 113–114, 146, 167, 177, 187, 213, 223, 236, 260, 285, 287, 293, 332, 335, 383, 438, 473, 484
- MigrateTable Spaces panel 76
- Migration panel 231
- multiple permissions 389
- MVS JCL 170
 - job 175
 - job stream 210

N

- NAME 22, 138, 162, 217–218, 294, 296, 371, 474
- Name field 113, 130, 218, 244, 349
- native SQL 81–82
 - SQL

- statement 90

- NB - Repair (NR) 298
- NORMAL COMPLETION 162, 294, 474
- NULL/Program name 150, 153, 336, 379, 445, 457
- Numparts 116–117, 182, 201

O

- O_1 column 325, 327
- object name 72, 181, 342, 411
- Object option 103, 113, 130, 299, 332, 467, 490
- object type 147, 238, 287, 411, 469, 489
- Operation Type 158, 250, 309
- operational library 20
- optimize DSNWZP 32, 62
- option 1 67, 112, 198, 230, 262, 283, 428, 472
- option 2 67, 190, 262, 295
- option 5 77, 255
- option line 83, 100, 112–113, 129, 145, 152, 238, 331–332, 420, 454, 462–463, 472, 480, 486
 - PDC command 477
 - SSID VA1B 100
- option x 103, 113, 130, 145, 299, 332, 349, 361, 467, 490
- option xC 342, 349, 369, 477, 491
- Optional job 121, 172, 253, 260, 428, 441
- Optional process 121, 172, 256, 260, 428, 441
- optional set 174
- OR 123, 138, 150, 153, 336, 340, 379, 445
- ORIGINAL 117, 182, 201, 203, 423

Output Report 254
OWNER 68, 89, 94, 98, 217, 238, 396, 398
owner 41, 67, 86, 115, 219, 238, 283, 332, 392

P

pages Segsz T L 288
PARMLIB 57
partition-by-growth (PBG) 109, 144, 149
partition-by-range (PBR) 109, 323
PDS See *prefix for data sets*
PENDING CHANGES 468
pending changes 466
period column 324
 End state 326
PF key 69
PGMNAME 217
PHONENO 188
PKGNAME 217
PKLIST 45
PLNNAME 217
PM27184 9
PM27186 9
PR 70–71
prefix for data sets (PDS) 121, 170, 236, 260, 428–429
 member 231, 266
 suffix 185
Primary Authid 396
primary command 4, 71, 116, 147, 238, 262, 284, 293, 467, 484
Primary key 149
PRINT 124
procedure name 103
product
 collection 14
 library 14
 new set 18
 parameter 65
 plan 14
program 5655-W34 408
Program parm 314
promote 214
PROMOTE process 238
PROMPT OFF 76
PROMPT ON 76

R

RC 123, 254
RDEF 115, 166
RE 107, 362
Recover TOCOPY (RT) 158, 160
RECOVERY ON command 90, 123, 205, 333, 344, 381, 450
Redbooks Web site 511
 Contact us xxvii
REFRESH 73, 139, 150, 153, 336, 347, 445
REGISTER 256
related object 126, 139, 149, 181, 284
remote subsystem 16, 68, 282
 Location name 68

Node name 68
 system catalog 283
Reorg 120, 159, 186, 441, 466
Repair norcvrpend 298
RESET 290, 293
Reset RI-FK flag 120, 178
Reset Source 243
Reset Target 243
Reset update 188, 438, 440
REVOKE 5, 76, 232, 267, 362
Revoke Table Privileges panel 368
REXX 14–15, 68–69, 85, 90, 215, 218, 267
 exit 217
RI 119, 176, 223, 427
Routine Version field 271
Row 1 77, 91–92, 113–114, 116, 146–148, 167, 188–189, 219, 241, 263, 282, 293, 309, 311–312, 332, 335, 362, 390, 418–419, 467, 492
ROW End 310
row permission 383
Rqd Oper 119, 176, 427
RUN 94, 232, 256, 267, 371, 441
Run CHECK Data 121, 186, 227
Run online 230
Run RUNSTATS 121, 210, 228
Run SQLID 74, 168, 225, 252, 392, 440, 489
RUNSTATS 14, 17, 115, 120, 186, 210, 225, 232, 253, 267, 359, 421
 information 59
 utility 17, 32, 62
 view 59
Runstats report (RR) 160
RW 421

S

SADBCLST library 46, 219
SADBSAMP library 39, 219
 ADBCATVT job 59
 ADBCHANG job 43
 ADBF2VB job 46
 ADBRUNSV job 60
 member ADBCHANG 219
same name 98, 266, 294, 341
SAMP 36
 ADBBIND member 45
 ADBCATVT member 59
 ADBCHANG member 43
 ADBCHKPT member 39
 ADBF2VB member 46, 57
 ADBRUNSV member 60
 appropriate member 61
SAMP library 39
 ADBCDCH member 43
SC 408
SCHEMA 151, 191, 217, 226, 233, 271, 280, 290
Schema Auth 91–92
Schema name 38, 175, 193, 444
scope objects (SO) 238
SEARCH 291–292, 370
Segment Size 114

- segmented table space 109–110
- segmented temporary table space 14
- SEGSIZE 110
- Sel Name 230, 263
 - VP Size 464
- SELECT MEM_ID 90
- select one 150, 153, 336, 379, 445
- select option
 - 2.4 406
 - TR 395
- Select prototyping 131, 142, 146, 167, 177, 187, 229, 285, 293, 332, 383, 438
- selection criterion 299
- selection panel 381
- separate SQL 358
- Sequence authorization 361, 491
- SET CURRENT SQLID 39, 43, 252, 256, 261, 333, 381, 476
- SFNAME 217
- SGNAME 217
- Show
 - command 232
 - object
 - R 181
 - V - View statement line command 233, 268
 - option 429
- Show Work Statement List panel 264
- SHRLEVEL REFERENCE 442, 466
- SMP/E 14
- SORTDEVT SYSDA 442
- SORTNUM 4 442
- source 8, 66, 125, 158, 169, 219, 227, 230, 236
- Specify Compare Source panel 244
- Specify Compare Target panel 246
- Specify Work Statement List panel 211
- SPUFI 6, 256
- SQ 106–107, 408, 467, 492
- SQL 81, 144, 151, 239, 282, 307, 414, 466, 484
 - body 82
 - DIAGNOSTIC INFORMATION 430, 480
 - procedure (SP) 82, 408, 480
 - statement 69, 82, 170, 215, 252, 260, 283, 295, 331, 383, 430, 485
 - ADB5178I RESULT 430–431
 - bind time 416
 - bind time info 412
- SQLID 5, 39–40, 74, 121–122, 168, 217, 225, 252, 260–261, 268, 309, 381, 440, 476, 489
- SSEMMDBB.SS12 DPT 469, 487
- ssid 68
- SST2UL1 227
- SST3CH 227
- SST4XF 227
- SST5DE 227
- ST 204, 294, 419
- standard selection criterion 103, 113, 145, 298, 332, 361, 467, 490
- Statement Execution 151, 155, 364, 420, 486
- Status change
 - date 159

- timestamp 159
- STLEC3B.ADMR 2 99
- stmts 371, 459, 486
- Stop table space 113–114, 116, 147–148, 297, 473
- Storage group 84, 223, 286, 307, 361, 487
 - DG 84, 307, 373
 - GRANT use 489
 - Image copy status (ICS) 127, 158, 478
 - P 291, 418, 495
- storage group 21, 38
- STPNAME 217
- subpanel 461
- Switch Catalog Copy 103, 113, 145, 332, 361, 490
- SYS_END column 316
 - Operation type 318
- SYS_STA column 311
 - col type 312
 - Operation type 315
- SYSADM.ALTMKPBR.APPL YJCL 137
- SYSADM.MAKEPBGA.APPL YJCL 122
- SYSADM.VA1A.IC.S000 1 442
- SYSIBM.SYSC OPY 158–159
 - DB0B Information 158
- SYSTABLESPACE panel 114, 127
- SYSTEM AUTHID
 - ADMR2 405
 - ADMR3 398
- System DBADM 359
 - new options 361
- system period
 - temporal table 306
- SYSTEM_TIME period 306
 - End column 323
 - following columns 310
 - transaction start ID 318

T

- T - Tables line command 167, 285, 418
- T - Tables, views, and aliases option 113, 467, 490
- T (type) column 242
- T option 145
- T type 332
- table list 293, 380, 383
 - PM command 390
- Table name 15, 86, 119, 146–147, 166, 286, 336, 379, 418, 443
- table schema 21, 119, 150–151, 176, 336, 379, 427
- table source 125, 170
- Table Space
 - panel 116, 182, 297–298
- Table space
 - BR 158
 - D 131, 142, 146, 167, 229, 285, 332, 335, 383, 438
 - Id 469
 - name 37, 146–147, 158, 181, 296, 469
 - SEGSIZE 37
 - status 115, 128
- table space 5, 23, 109, 127, 130, 144, 146, 166, 213, 223, 238, 283, 286, 417, 443, 466, 487
 - conversion 136

- DBAADMR2.TPAA DMR2 125
- detail information 114
- level 8, 296
- MAXPARTITIONS 144
- modification 202
- new type 109
- Oper column 135
- organization 109
- Original Definition 470
- other type 110
- partition 265, 455
- Recovery details 159
- TDTEAM76.TD76 TS14 160–161
- TPAADMR2 112–114, 130–131
- type 109, 113
- Table Space maintenance recommendation option 494
- TABLE SPACE PARTITIONS 264
- Table Spaces panel 113, 148, 473
- TABLE TD76TB03_ACT 228
- table type 144, 346, 353
- Tables panel 119
- tablespace source 211
- tablespace SSEMMDBB.SS12 485
- TABLESPACE statement 110
- TABLESPACE TD76TS07 228
- target 5, 7–8, 14, 19, 98, 125, 169, 214–215, 236, 283–284
- target DB2
 - system 100
- TBADD304 table 131
 - partition key 132
- TBNAME 217, 292, 296
- TC command 301
- TEAM76.TD76 TB01_DEPT 170
- TEAM76.TD76 TB02_EMP 170, 211
- TEAM76.TD76 TB06_EMPPROJACT 203
- TEAM89.TD89 TB02_EMP 423
- temporal table 305
- TGNAME 217
- time column 306
- TIME ZONE 314, 439
 - TIMESTAMP 455
 - TIMESTAMP length 458
- timestamp 306, 439
- timestamp value 350
- TRIGGER 368, 489
- trusted context 358, 373
 - additional attributes 398
 - customer database 398
- TS 166, 408
- TS owner 128
- TSBPNAME 217
- TSNAME 217–218, 296, 301
- TSO command 57
- TSSGNAME 217
- TST1CR 227
- TST3CK 227
- TST4RS 228
- TST6RB 228
- TT 332

- Type name 314, 439
- Type schema 314, 439

U

- U - Update line command 220, 241–242, 309
- U Lock for RR or RS parameter 360
- UDFNAME 217
- UDTNAME 217
- UNIQUE constraint 150, 153, 336, 379, 445
- UNIQUE INDEX
 - TD76XA01 228
 - TD76XA02 228
 - TD76XA03 228
 - TD76XA06 228
 - TD76XA14 228
 - TD76XB03 228
- UNIT SYSDA 442
- universal table space (UTS) 109–110, 150, 159
- Unload method 120–121, 172, 184–185, 225, 255, 260
- Unload option 160
- UPD TIMESTAMP 314, 439
- update RUNSTATS (UR) 59
- Use masking for batch migrate option 225
- user exit 215
- user ID 37, 262, 373, 429
- user restart point 275
- User-defined type 314
 - name 439
 - schema 317, 439
- user-specified restart 270

V

- V - Generate job to extract version file from source only option 243
- V - Validate line command 230, 263
- V - Views line command 167, 177, 229, 285, 383, 429
- V - Views option 103, 113, 145, 332, 467, 490
- V10152010 package 99
- VA1A
 - ALTER Table panel 438–440
 - Alter Table panel 336, 389
 - Alter Tables panel 126
 - Interpretation of an Object in SYSCONTROLS panel 382
 - Interpretation of an Object in SYSTABLESPACE panel 128
 - System Catalog panel 130, 332, 342, 361
 - Tables, Views, and Aliases panel 131, 332, 383, 438
- Validate 252
- Validate category 411
- VALUES 117, 133, 350
- variable-block library 25
- VCATNAME 217
- VE 106, 241, 492
- Version column 91
- Version files 237
- version identifier 84
- Version Scope 235, 237, 254
- version scope 16, 214, 238

- new 238
- View DDL 197
- VIEW TEAM76 191
- views on views (VV) 286

W

- work data 172, 175, 184
- work statement list (WSL) 5, 8, 19, 37, 83, 112, 120, 123, 151, 155–156, 170, 185, 214–215, 222, 229–231, 239, 253, 259–261, 264, 279, 283, 331, 364, 420, 429, 468, 484–486
 - data 261
 - data set 261–262
 - DB2 Object Comparison Tool 262
 - DSN 155–156, 261, 371, 377, 454, 459, 471, 480
 - DSN name 262
 - execution 430
 - library 175, 211, 230–231, 262–263, 429
 - library member 481
 - library name 211, 429
 - list 231
 - masking 229
 - member 170, 175, 184, 429, 454, 472, 481
 - name 155–156, 261–262, 364, 371, 420, 454, 468, 471, 486
- Work Statement List Library panel 263, 266–268
- Work Statement List Options panel 267, 429
- Worklist information 121, 172, 251, 260, 428
- Worklist name 121–122, 172, 184, 209, 225, 251, 260, 428, 441
- WSL See *worklist statement list*

X

- xC 103, 113, 130, 145, 299–300, 332, 361, 467, 490
- XML Count 264
- XML Modifier 444
 - XML column 444
- xxxxxS1 job name 227
- xxxxxSE job name 227
- xxxxxT1 job name 227

Y

- YNAMNBR 254



Redbooks

Managing IBM DB2 10 for z/OS Using the IBM DB2 Administration Tool for z/OS Version 10

(1.0" spine)

0.875" <-> 1.498"

460 <-> 788 pages



Redbooks®

Managing IBM DB2 10 for z/OS Using the IBM DB2 Administration Tool for z/OS Version 10

**Use with IBM DB2 10
for z/OS from Day 1**

**Simplify the database
administration
functions**

**Adapt your data with
the new change
capabilities**

Today's business environment has increased in the complexity and rate of change that a database administrator must control. The ability to respond quickly to a changing environment is constantly challenged by the explosion of data growth combined with a decline in an experienced work staff.

The IBM DB2 Administration Tool for z/OS Version 10 helps you become productive from Day 1 with DB2 10 for z/OS by using performance savings right away, lowering the CPU costs while reducing the batch window. Users experience higher data availability by easily managing online schema changes, and include additional columns to indexes to exploit index-only access

Customers are able to experience higher data availability through simplified recovery operations:

- ▶ Access new functionality in DB2 10 for z/OS to lower costs and improve efficiency both before, during, and after the DB2 migration process.
- ▶ Maximize the performance of your key DB2 business applications to speed their deployment in DB2 10 for z/OS.
- ▶ Improve the productivity and efficiency of your staff when DB2 10 for z/OS is running.

This IBM Redbooks publication highlights the data administration enhancements introduced by DB2 Administration Tool for z/OS Version 10 by providing scenarios of their use with the new functions provided by DB2 10 for z/OS.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks